Attacks on DNS

Cryptography in DNS

Secure design and coding for DNS

D. J. Bernstein
University of Illinois at Chicago

http://cr.yp.to
/talks.html#2009.03.02
/talks.html#2009.03.03
/talks.html#2009.03.04

1996: qmail 0.70.
1997: qmail 1.00.
1998: qmail 1.03.

Attacks on DNS

Cryptography in DNS

Secure design and coding for DNS

D. J. Bernstein
University of Illinois at Chicago

http://cr.yp.to
/talks.html#2009.03.02
/talks.html#2009.03.03
/talks.html#2009.03.04

1996: qmail 0.70.
1997: qmail 1.00.
1998: qmail 1.03.
1999: djbdns (dnscache) 0.60.
2000: djbdns (dnscache) 1.00.
2001: djbdns 1.05.

Attacks on DNS

Cryptography in DNS

Secure design and coding for DNS

D. J. Bernstein
University of Illinois at Chicago

http://cr.yp.to
/talks.html#2009.03.02
/talks.html#2009.03.03
/talks.html#2009.03.04

1996: qmail 0.70.
1997: qmail 1.00.
1998: qmail 1.03.
1999: djbdns (dnscache) 0.60.
2000: djbdns (dnscache) 1.00.
2001: djbdns 1.05.

2007: “Some thoughts on security after ten years of qmail 1.0.”

Attacks on DNS

Cryptography in DNS

Secure design and coding for DNS

D. J. Bernstein
University of Illinois at Chicago

http://cr.yp.to
/talks.html#2009.03.02
/talks.html#2009.03.03
/talks.html#2009.03.04

1996: qmail 0.70.
1997: qmail 1.00.
1998: qmail 1.03.
1999: djbdns (dnscache) 0.60.
2000: djbdns (dnscache) 1.00.
2001: djbdns 1.05.

2007: "Some thoughts on security after ten years of qmail 1.0."

$>$ 1000000 of the Internet's SMTP servers are running qmail.
$>$ 4000000 of the Internet's second-level `*.com` names are published by djbdns.

Attacks on DNS

Cryptography in DNS

Secure design and coding for DNS

D. J. Bernstein
University of Illinois at Chicago

http://cr.yp.to
/talks.html#2009.03.02
/talks.html#2009.03.03
/talks.html#2009.03.04

1996: qmail 0.70.
1997: qmail 1.00.
1998: qmail 1.03.
1999: djbdns (dnscache) 0.60.
2000: djbdns (dnscache) 1.00.
2001: djbdns 1.05.

2007: "Some thoughts on security after ten years of qmail 1.0."

$>$ 1000000 of the Internet's SMTP servers are running qmail.
$>$ 4000000 of the Internet's second-level `*.com` names are published by djbdns.

No emergency upgrades, ever.

s on DNS

graphy in DNS

design and coding for DNS

Bernstein

sity of Illinois at Chicago

//cr.yp.to

s.html#2009.03.02

s.html#2009.03.03

s.html#2009.03.04

1996: qmail 0.70.

1997: qmail 1.00.

1998: qmail 1.03.

1999: djbdns (dnscache) 0.60.

2000: djbdns (dnscache) 1.00.

2001: djbdns 1.05.

2007: "Some thoughts on security after ten years of qmail 1.0."

$>$ 1000000 of the Internet's SMTP servers are running qmail.

$>$ 4000000 of the Internet's second-level `*.com` names are published by djbdns.

No emergency upgrades, ever.

Some D

"CVE-20 overflow function before 1. execute TXT rec

"CVE-20 overflow split.c in with the remote a code via

"CVE-20 Cisco Un 7960, an firmware execute response

DNS

d coding for DNS

ois at Chicago

to

009.03.02

009.03.03

009.03.04

1996: qmail 0.70.

1997: qmail 1.00.

1998: qmail 1.03.

1999: djbdns (dnscache) 0.60.

2000: djbdns (dnscache) 1.00.

2001: djbdns 1.05.

2007: "Some thoughts on security after ten years of qmail 1.0."

$>$ 1000000 of the Internet's SMTP servers are running qmail.

$>$ 4000000 of the Internet's second-level `*.com` names are published by djbdns.

No emergency upgrades, ever.

Some DNS buffe

"CVE-2008-2469: H
overflow in the SPF_
function in Spf_dns_r
before 1.2.8 allows r
execute arbitrary cod
TXT record with a r

"CVE-2008-2357: St
overflow in the split_
split.c in mtr before
with the -p (aka –sp
remote attackers to
code via a crafted D

"CVE-2008-0530: Bu
Cisco Unified IP Pho
7960, and 7960G run
firmware might allow
execute arbitrary cod
response."

for DNS

cago

2
3
4

1996: qmail 0.70.
1997: qmail 1.00.
1998: qmail 1.03.
1999: djbdns (dnscache) 0.60.
2000: djbdns (dnscache) 1.00.
2001: djbdns 1.05.

2007: "Some thoughts on security after ten years of qmail 1.0."

$>$ 1000000 of the Internet's SMTP servers are running qmail.
$>$ 4000000 of the Internet's second-level `*.com` names are published by djbdns.

No emergency upgrades, ever.

Some DNS buffer overflow

"CVE-2008-2469: Heap-based bu
overflow in the SPF_dns_resolv_lo
function in Spf_dns_resolv.c in lib
before 1.2.8 allows remote attack
execute arbitrary code via a long
TXT record with a modified leng

"CVE-2008-2357: Stack-based b
overflow in the split_redraw funct
split.c in mtr before 0.73, when
with the -p (aka –split) option, a
remote attackers to execute arbi
code via a crafted DNS PTR rec

"CVE-2008-0530: Buffer overflow
Cisco Unified IP Phone 7940, 79
7960, and 7960G running SCCP
firmware might allow remote att
execute arbitrary code via a craf
response."

1996: qmail 0.70.

1997: qmail 1.00.

1998: qmail 1.03.

1999: djbdns (dnscache) 0.60.

2000: djbdns (dnscache) 1.00.

2001: djbdns 1.05.

2007: "Some thoughts on security after ten years of qmail 1.0."

$>$ 1000000 of the Internet's SMTP servers are running qmail.

$>$ 4000000 of the Internet's second-level `*.com` names are published by djbdns.

No emergency upgrades, ever.

Some DNS buffer overflows

"CVE-2008-2469: Heap-based buffer overflow in the SPF_dns_resolv_lookup function in Spf_dns_resolv.c in libspf2 before 1.2.8 allows remote attackers to execute arbitrary code via a long DNS TXT record with a modified length field."

"CVE-2008-2357: Stack-based buffer overflow in the split_redraw function in split.c in mtr before 0.73, when invoked with the -p (aka –split) option, allows remote attackers to execute arbitrary code via a crafted DNS PTR record."

"CVE-2008-0530: Buffer overflow in Cisco Unified IP Phone 7940, 7940G, 7960, and 7960G running SCCP and SIP firmware might allow remote attackers to execute arbitrary code via a crafted DNS response."

qmail 0.70.
qmail 1.00.
qmail 1.03.
djbdns (dnscache) 0.60.
djbdns (dnscache) 1.00.
djbdns 1.05.

"Some thoughts on security
en years of qmail 1.0."

0000 of the Internet's
servers are running qmail.
0000 of the Internet's
-level *.com names
blished by djbdns.

ergency upgrades, ever.

Some DNS buffer overflows

"CVE-2008-2469: Heap-based buffer overflow in the SPF_dns_resolv_lookup function in Spf_dns_resolv.c in libspf2 before 1.2.8 allows remote attackers to execute arbitrary code via a long DNS TXT record with a modified length field."

"CVE-2008-2357: Stack-based buffer overflow in the split_redraw function in split.c in mtr before 0.73, when invoked with the -p (aka –split) option, allows remote attackers to execute arbitrary code via a crafted DNS PTR record."

"CVE-2008-0530: Buffer overflow in Cisco Unified IP Phone 7940, 7940G, 7960, and 7960G running SCCP and SIP firmware might allow remote attackers to execute arbitrary code via a crafted DNS response."

"CVE-20
inet_netw
BIND 9.
FreeBSD
allows co
cause a
possibly
input tha

"CVE-20
asnsp.dll
allows re
of servic
arbitrary
query."

"CVE-20
overflows
attackers
(daemon
arbitrary
triggers

0.

0.

3.

nscache) 0.60.

nscache) 1.00.

05.

oughts on security

f qmail 1.0."

e Internet's

e running qmail.

e Internet's

om names

djbdns.

pgrades, ever.

Some DNS buffer overflows

"CVE-2008-2469: Heap-based buffer overflow in the SPF_dns_resolv_lookup function in Spf_dns_resolv.c in libspf2 before 1.2.8 allows remote attackers to execute arbitrary code via a long DNS TXT record with a modified length field."

"CVE-2008-2357: Stack-based buffer overflow in the split_redraw function in split.c in mtr before 0.73, when invoked with the -p (aka –split) option, allows remote attackers to execute arbitrary code via a crafted DNS PTR record."

"CVE-2008-0530: Buffer overflow in Cisco Unified IP Phone 7940, 7940G, 7960, and 7960G running SCCP and SIP firmware might allow remote attackers to execute arbitrary code via a crafted DNS response."

"CVE-2008-0122: O
inet_network function
BIND 9.4.2 and earli
FreeBSD 6.2 through
allows context-depen
cause a denial of ser
possibly execute arbi
input that triggers m

"CVE-2007-2434: Bu
asnsp.dll in Aventail
allows remote attack
of service (applicatio
arbitrary code via a
query."

"CVE-2007-2362: M
overflows in MyDNS
attackers to (1) caus
(daemon crash) and
arbitrary code via a
triggers a heap-based

.60.
.00.

security
0."

s
qmail.
s

ver.

Some DNS buffer overflows

"CVE-2008-2469: Heap-based buffer overflow in the SPF_dns_resolv_lookup function in Spf_dns_resolv.c in libspf2 before 1.2.8 allows remote attackers to execute arbitrary code via a long DNS TXT record with a modified length field."

"CVE-2008-2357: Stack-based buffer overflow in the split_redraw function in split.c in mtr before 0.73, when invoked with the -p (aka –split) option, allows remote attackers to execute arbitrary code via a crafted DNS PTR record."

"CVE-2008-0530: Buffer overflow in Cisco Unified IP Phone 7940, 7940G, 7960, and 7960G running SCCP and SIP firmware might allow remote attackers to execute arbitrary code via a crafted DNS response."

"CVE-2008-0122: Off-by-one err
inet_network function in libbind i
BIND 9.4.2 and earlier, as used
FreeBSD 6.2 through 7.0-PRERE
allows context-dependent attacke
cause a denial of service (crash)
possibly execute arbitrary code v
input that triggers memory corru

"CVE-2007-2434: Buffer overflow
asnsp.dll in Aventail Connect 4.1
allows remote attackers to cause
of service (application crash) or
arbitrary code via a malformed D
query."

"CVE-2007-2362: Multiple buffe
overflows in MyDNS 1.1.0 allow
attackers to (1) cause a denial o
(daemon crash) and possibly exe
arbitrary code via a certain upda
triggers a heap-based buffer over

Some DNS buffer overflows

"CVE-2008-2469: Heap-based buffer overflow in the SPF_dns_resolv_lookup function in Spf_dns_resolv.c in libspf2 before 1.2.8 allows remote attackers to execute arbitrary code via a long DNS TXT record with a modified length field."

"CVE-2008-2357: Stack-based buffer overflow in the split_redraw function in split.c in mtr before 0.73, when invoked with the -p (aka –split) option, allows remote attackers to execute arbitrary code via a crafted DNS PTR record."

"CVE-2008-0530: Buffer overflow in Cisco Unified IP Phone 7940, 7940G, 7960, and 7960G running SCCP and SIP firmware might allow remote attackers to execute arbitrary code via a crafted DNS response."

"CVE-2008-0122: Off-by-one error in the inet_network function in libbind in ISC BIND 9.4.2 and earlier, as used in libc in FreeBSD 6.2 through 7.0-PRERELEASE, allows context-dependent attackers to cause a denial of service (crash) and possibly execute arbitrary code via crafted input that triggers memory corruption."

"CVE-2007-2434: Buffer overflow in asnsp.dll in Aventail Connect 4.1.2.13 allows remote attackers to cause a denial of service (application crash) or execute arbitrary code via a malformed DNS query."

"CVE-2007-2362: Multiple buffer overflows in MyDNS 1.1.0 allow remote attackers to (1) cause a denial of service (daemon crash) and possibly execute arbitrary code via a certain update, which triggers a heap-based buffer overflow

DNS buffer overflows

008-2469: Heap-based buffer
in the SPF_dns_resolv_lookup
in Spf_dns_resolv.c in libspf2
.2.8 allows remote attackers to
arbitrary code via a long DNS
cord with a modified length field."

008-2357: Stack-based buffer
in the split_redraw function in
mtr before 0.73, when invoked
-p (aka –split) option, allows
attackers to execute arbitrary
a crafted DNS PTR record."

008-0530: Buffer overflow in
nified IP Phone 7940, 7940G,
nd 7960G running SCCP and SIP
might allow remote attackers to
arbitrary code via a crafted DNS
."

"CVE-2008-0122: Off-by-one error in the inet_network function in libbind in ISC BIND 9.4.2 and earlier, as used in libc in FreeBSD 6.2 through 7.0-PRERELEASE, allows context-dependent attackers to cause a denial of service (crash) and possibly execute arbitrary code via crafted input that triggers memory corruption."

"CVE-2007-2434: Buffer overflow in asnsp.dll in Aventail Connect 4.1.2.13 allows remote attackers to cause a denial of service (application crash) or execute arbitrary code via a malformed DNS query."

"CVE-2007-2362: Multiple buffer overflows in MyDNS 1.1.0 allow remote attackers to (1) cause a denial of service (daemon crash) and possibly execute arbitrary code via a certain update, which triggers a heap-based buffer overflow

in updat
service (
vectors t
based bu

"CVE-20
overflow
allows re
arbitrary

"CVE-20
overflow
function
allows re
arbitrary
packet t

"CVE-20
overflow
Domain
Service i
SP 4, Se

r overflows

eap-based buffer
_dns_resolv_lookup
resolv.c in libspf2
emote attackers to
de via a long DNS
modified length field."

tack-based buffer
redraw function in
0.73, when invoked
lit) option, allows
execute arbitrary
NS PTR record."

uffer overflow in
one 7940, 7940G,
nning SCCP and SIP
remote attackers to
de via a crafted DNS

"CVE-2008-0122: Off-by-one error in the inet_network function in libbind in ISC BIND 9.4.2 and earlier, as used in libc in FreeBSD 6.2 through 7.0-PRERELEASE, allows context-dependent attackers to cause a denial of service (crash) and possibly execute arbitrary code via crafted input that triggers memory corruption."

"CVE-2007-2434: Buffer overflow in asnsp.dll in Aventail Connect 4.1.2.13 allows remote attackers to cause a denial of service (application crash) or execute arbitrary code via a malformed DNS query."

"CVE-2007-2362: Multiple buffer overflows in MyDNS 1.1.0 allow remote attackers to (1) cause a denial of service (daemon crash) and possibly execute arbitrary code via a certain update, which triggers a heap-based buffer overflow

in update.c; and (2)
service (daemon cras
vectors that trigger a
based buffer overflow

"CVE-2007-2187: St
overflow in eXtremai
allows remote attack
arbitrary code via a

"CVE-2007-1866: St
overflow in the dns_d
function in dns_deco
allows remote attack
arbitrary code by ser
packet to port 53/uc

"CVE-2007-1748: St
overflow in the RPC
Domain Name Syste
Service in Microsoft
SP 4, Server 2003 S

s

uffer
ookup
pspf2
kers to
g DNS
gth field."

uffer
tion in
invoked
allows
trary
cord."

w in
040G,
and SIP
ackers to
ted DNS

"CVE-2008-0122: Off-by-one error in the inet_network function in libbind in ISC BIND 9.4.2 and earlier, as used in libc in FreeBSD 6.2 through 7.0-PRERELEASE, allows context-dependent attackers to cause a denial of service (crash) and possibly execute arbitrary code via crafted input that triggers memory corruption."

"CVE-2007-2434: Buffer overflow in asnsp.dll in Aventail Connect 4.1.2.13 allows remote attackers to cause a denial of service (application crash) or execute arbitrary code via a malformed DNS query."

"CVE-2007-2362: Multiple buffer overflows in MyDNS 1.1.0 allow remote attackers to (1) cause a denial of service (daemon crash) and possibly execute arbitrary code via a certain update, which triggers a heap-based buffer overflow

in update.c; and (2) cause a den
service (daemon crash) via unspe
vectors that trigger an off-by-one
based buffer overflow in update.c

"CVE-2007-2187: Stack-based b
overflow in eXtremail 2.1.1 and
allows remote attackers to execu
arbitrary code via a long DNS re

"CVE-2007-1866: Stack-based b
overflow in the dns_decode_revers
function in dns_decode.c in dprox
allows remote attackers to execu
arbitrary code by sending a craft
packet to port 53/udp."

"CVE-2007-1748: Stack-based b
overflow in the RPC interface in
Domain Name System (DNS) Se
Service in Microsoft Windows 20
SP 4, Server 2003 SP 1, and Se

"CVE-2008-0122: Off-by-one error in the inet_network function in libbind in ISC BIND 9.4.2 and earlier, as used in libc in FreeBSD 6.2 through 7.0-PRERELEASE, allows context-dependent attackers to cause a denial of service (crash) and possibly execute arbitrary code via crafted input that triggers memory corruption."

"CVE-2007-2434: Buffer overflow in asnsp.dll in Aventail Connect 4.1.2.13 allows remote attackers to cause a denial of service (application crash) or execute arbitrary code via a malformed DNS query."

"CVE-2007-2362: Multiple buffer overflows in MyDNS 1.1.0 allow remote attackers to (1) cause a denial of service (daemon crash) and possibly execute arbitrary code via a certain update, which triggers a heap-based buffer overflow in update.c; and (2) cause a denial of service (daemon crash) via unspecified vectors that trigger an off-by-one stack-based buffer overflow in update.c."

"CVE-2007-2187: Stack-based buffer overflow in eXtremail 2.1.1 and earlier allows remote attackers to execute arbitrary code via a long DNS response."

"CVE-2007-1866: Stack-based buffer overflow in the dns_decode_reverse_name function in dns_decode.c in dproxy-nexgen allows remote attackers to execute arbitrary code by sending a crafted packet to port 53/udp."

"CVE-2007-1748: Stack-based buffer overflow in the RPC interface in the Domain Name System (DNS) Server Service in Microsoft Windows 2000 Server SP 4, Server 2003 SP 1, and Server 2003

008-0122: Off-by-one error in the work function in libbind in ISC 4.2 and earlier, as used in libc in 6.2 through 7.0-PRERELEASE, ontext-dependent attackers to denial of service (crash) and execute arbitrary code via crafted at triggers memory corruption."

007-2434: Buffer overflow in in Aventail Connect 4.1.2.13 emote attackers to cause a denial e (application crash) or execute code via a malformed DNS

007-2362: Multiple buffer s in MyDNS 1.1.0 allow remote s to (1) cause a denial of service crash) and possibly execute code via a certain update, which a heap-based buffer overflow

in update.c; and (2) cause a denial of service (daemon crash) via unspecified vectors that trigger an off-by-one stack-based buffer overflow in update.c."

"CVE-2007-2187: Stack-based buffer overflow in eXtremail 2.1.1 and earlier allows remote attackers to execute arbitrary code via a long DNS response."

"CVE-2007-1866: Stack-based buffer overflow in the dns_decode_reverse_name function in dns_decode.c in dproxy-nexgen allows remote attackers to execute arbitrary code by sending a crafted packet to port 53/udp."

"CVE-2007-1748: Stack-based buffer overflow in the RPC interface in the Domain Name System (DNS) Server Service in Microsoft Windows 2000 Server SP 4, Server 2003 SP 1, and Server 2003

SP 2 all arbitrary containi by escap

"CVE-20 overflow through execute query pa

"CVE-20 overflow iodine 0. execute response

"CVE-20 PowerDN might al arbitrary DNS que

ff-by-one error in the
n in libbind in ISC
ier, as used in libc in
n 7.0-PRERELEASE,
dent attackers to
rvice (crash) and
trary code via crafted
nemory corruption."

uffer overflow in
Connect 4.1.2.13
kers to cause a denial
n crash) or execute
malformed DNS

ultiple buffer
1.1.0 allow remote
se a denial of service
possibly execute
certain update, which
d buffer overflow

in update.c; and (2) cause a denial of service (daemon crash) via unspecified vectors that trigger an off-by-one stack-based buffer overflow in update.c."

"CVE-2007-2187: Stack-based buffer overflow in eXtremail 2.1.1 and earlier allows remote attackers to execute arbitrary code via a long DNS response."

"CVE-2007-1866: Stack-based buffer overflow in the dns_decode_reverse_name function in dns_decode.c in dproxy-nexgen allows remote attackers to execute arbitrary code by sending a crafted packet to port 53/udp."

"CVE-2007-1748: Stack-based buffer overflow in the RPC interface in the Domain Name System (DNS) Server Service in Microsoft Windows 2000 Server SP 4, Server 2003 SP 1, and Server 2003

SP 2 allows remote
arbitrary code via a
containing character
by escape sequences.

"CVE-2007-1465: St
overflow in dproxy.c
through 0.5 allows re
execute arbitrary cod
query packet to UDP

"CVE-2006-5781: St
overflow in the hand
iodine 0.3.2 allows re
execute arbitrary cod
response."

"CVE-2006-4251: Bu
PowerDNS Recursor
might allow remote a
arbitrary code via a
DNS query that prev

or in the
n ISC
in libc in
ELEASE,
ers to
and
ia crafted
uption."

w in
..2.13
a denial
execute
DNS

r
remote
of service
ecute
ate, which
rflow

in update.c; and (2) cause a denial of service (daemon crash) via unspecified vectors that trigger an off-by-one stack-based buffer overflow in update.c."

"CVE-2007-2187: Stack-based buffer overflow in eXtremail 2.1.1 and earlier allows remote attackers to execute arbitrary code via a long DNS response."

"CVE-2007-1866: Stack-based buffer overflow in the dns_decode_reverse_name function in dns_decode.c in dproxy-nexgen allows remote attackers to execute arbitrary code by sending a crafted packet to port 53/udp."

"CVE-2007-1748: Stack-based buffer overflow in the RPC interface in the Domain Name System (DNS) Server Service in Microsoft Windows 2000 Server SP 4, Server 2003 SP 1, and Server 2003

SP 2 allows remote attackers to
arbitrary code via a long zone na
containing character constants re
by escape sequences."

"CVE-2007-1465: Stack-based b
overflow in dproxy.c for dproxy 0
through 0.5 allows remote attack
execute arbitrary code via a long
query packet to UDP port 53."

"CVE-2006-5781: Stack-based b
overflow in the handshake functi
iodine 0.3.2 allows remote attack
execute arbitrary code via a craf
response."

"CVE-2006-4251: Buffer overflow
PowerDNS Recursor 3.1.3 and ea
might allow remote attackers to
arbitrary code via a malformed T
DNS query that prevents Recurs

in update.c; and (2) cause a denial of service (daemon crash) via unspecified vectors that trigger an off-by-one stack-based buffer overflow in update.c."

"CVE-2007-2187: Stack-based buffer overflow in eXtremail 2.1.1 and earlier allows remote attackers to execute arbitrary code via a long DNS response."

"CVE-2007-1866: Stack-based buffer overflow in the dns_decode_reverse_name function in dns_decode.c in dproxy-nexgen allows remote attackers to execute arbitrary code by sending a crafted packet to port 53/udp."

"CVE-2007-1748: Stack-based buffer overflow in the RPC interface in the Domain Name System (DNS) Server Service in Microsoft Windows 2000 Server SP 4, Server 2003 SP 1, and Server 2003 SP 2 allows remote attackers to execute arbitrary code via a long zone name containing character constants represented by escape sequences."

"CVE-2007-1465: Stack-based buffer overflow in dproxy.c for dproxy 0.1 through 0.5 allows remote attackers to execute arbitrary code via a long DNS query packet to UDP port 53."

"CVE-2006-5781: Stack-based buffer overflow in the handshake function in iodine 0.3.2 allows remote attackers to execute arbitrary code via a crafted DNS response."

"CVE-2006-4251: Buffer overflow in PowerDNS Recursor 3.1.3 and earlier might allow remote attackers to execute arbitrary code via a malformed TCP DNS query that prevents Recursor from

e.c; and (2) cause a denial of daemon crash) via unspecified that trigger an off-by-one stack- uffer overflow in update.c."

007-2187: Stack-based buffer in eXtremail 2.1.1 and earlier emote attackers to execute code via a long DNS response."

007-1866: Stack-based buffer in the dns_decode_reverse_name in dns_decode.c in dproxy-nexgen emote attackers to execute code by sending a crafted o port 53/udp."

007-1748: Stack-based buffer in the RPC interface in the Name System (DNS) Server n Microsoft Windows 2000 Server erver 2003 SP 1, and Server 2003 SP 2 allows remote attackers to execute arbitrary code via a long zone name containing character constants represented by escape sequences."

"CVE-2007-1465: Stack-based buffer overflow in dproxy.c for dproxy 0.1 through 0.5 allows remote attackers to execute arbitrary code via a long DNS query packet to UDP port 53."

"CVE-2006-5781: Stack-based buffer overflow in the handshake function in iodine 0.3.2 allows remote attackers to execute arbitrary code via a crafted DNS response."

"CVE-2006-4251: Buffer overflow in PowerDNS Recursor 3.1.3 and earlier might allow remote attackers to execute arbitrary code via a malformed TCP DNS query that prevents Recursor from

properly length."

"CVE-20 DNS Cli 2000 SP 2003 SP execute record re implies t are mult vulnerab based bu response response and (3) HINFO,

"CVE-20 Domain before 2. execute

cause a denial of
sh) via unspecified
an off-by-one stack-
w in update.c."

tack-based buffer
l 2.1.1 and earlier
kers to execute
long DNS response."

tack-based buffer
decode_reverse_name
de.c in dproxy-nexgen
kers to execute
nding a crafted
dp."

tack-based buffer
interface in the
m (DNS) Server
Windows 2000 Server
P 1, and Server 2003

SP 2 allows remote attackers to execute arbitrary code via a long zone name containing character constants represented by escape sequences."

"CVE-2007-1465: Stack-based buffer overflow in dproxy.c for dproxy 0.1 through 0.5 allows remote attackers to execute arbitrary code via a long DNS query packet to UDP port 53."

"CVE-2006-5781: Stack-based buffer overflow in the handshake function in iodine 0.3.2 allows remote attackers to execute arbitrary code via a crafted DNS response."

"CVE-2006-4251: Buffer overflow in PowerDNS Recursor 3.1.3 and earlier might allow remote attackers to execute arbitrary code via a malformed TCP DNS query that prevents Recursor from

properly calculating
length."

"CVE-2006-3441: Bu
DNS Client service i
2000 SP4, XP SP1 a
2003 SP1 allows rem
execute arbitrary cod
record response. NO
implies that there is
are multiple vectors,
vulnerabilities, relate
based buffer overflow
response to the clien
response with malfor
and (3) a length mis
HINFO, X25, and IS

"CVE-2005-2315: Bu
Domain Name Relay
before 2.19.1 allows
execute arbitrary cod

nial of
ecified
e stack-
c."

uffer
earlier
te
esponse."

uffer
se_name
xy-nexgen
te
ted

uffer
the
erver
000 Server
rver 2003

SP 2 allows remote attackers to execute arbitrary code via a long zone name containing character constants represented by escape sequences."

"CVE-2007-1465: Stack-based buffer overflow in dproxy.c for dproxy 0.1 through 0.5 allows remote attackers to execute arbitrary code via a long DNS query packet to UDP port 53."

"CVE-2006-5781: Stack-based buffer overflow in the handshake function in iodine 0.3.2 allows remote attackers to execute arbitrary code via a crafted DNS response."

"CVE-2006-4251: Buffer overflow in PowerDNS Recursor 3.1.3 and earlier might allow remote attackers to execute arbitrary code via a malformed TCP DNS query that prevents Recursor from

properly calculating the TCP DN
length."

"CVE-2006-3441: Buffer overflow
DNS Client service in Microsoft
2000 SP4, XP SP1 and SP2, an
2003 SP1 allows remote attacker
execute arbitrary code via a craf
record response. NOTE: while M
implies that there is a single issu
are multiple vectors, and likely m
vulnerabilities, related to (1) a h
based buffer overflow in a DNS
response to the client, (2) a DN
response with malformed ATMA
and (3) a length miscalculation i
HINFO, X25, and ISDN records."

"CVE-2005-2315: Buffer overflow
Domain Name Relay Daemon (D
before 2.19.1 allows remote attac
execute arbitrary code via a large

SP 2 allows remote attackers to execute arbitrary code via a long zone name containing character constants represented by escape sequences."

"CVE-2007-1465: Stack-based buffer overflow in dproxy.c for dproxy 0.1 through 0.5 allows remote attackers to execute arbitrary code via a long DNS query packet to UDP port 53."

"CVE-2006-5781: Stack-based buffer overflow in the handshake function in iodine 0.3.2 allows remote attackers to execute arbitrary code via a crafted DNS response."

"CVE-2006-4251: Buffer overflow in PowerDNS Recursor 3.1.3 and earlier might allow remote attackers to execute arbitrary code via a malformed TCP DNS query that prevents Recursor from properly calculating the TCP DNS query length."

"CVE-2006-3441: Buffer overflow in the DNS Client service in Microsoft Windows 2000 SP4, XP SP1 and SP2, and Server 2003 SP1 allows remote attackers to execute arbitrary code via a crafted record response. NOTE: while MS06-041 implies that there is a single issue, there are multiple vectors, and likely multiple vulnerabilities, related to (1) a heap-based buffer overflow in a DNS server response to the client, (2) a DNS server response with malformed ATMA records, and (3) a length miscalculation in TXT, HINFO, X25, and ISDN records."

"CVE-2005-2315: Buffer overflow in Domain Name Relay Daemon (DNRD) before 2.19.1 allows remote attackers to execute arbitrary code via a large number

ows remote attackers to execute
code via a long zone name
ng character constants represented
e sequences."

007-1465: Stack-based buffer
in dproxy.c for dproxy 0.1
0.5 allows remote attackers to
arbitrary code via a long DNS
acket to UDP port 53."

006-5781: Stack-based buffer
in the handshake function in
3.2 allows remote attackers to
arbitrary code via a crafted DNS
."

006-4251: Buffer overflow in
NS Recursor 3.1.3 and earlier
low remote attackers to execute
code via a malformed TCP
ery that prevents Recursor from

properly calculating the TCP DNS query length."

"CVE-2006-3441: Buffer overflow in the DNS Client service in Microsoft Windows 2000 SP4, XP SP1 and SP2, and Server 2003 SP1 allows remote attackers to execute arbitrary code via a crafted record response. NOTE: while MS06-041 implies that there is a single issue, there are multiple vectors, and likely multiple vulnerabilities, related to (1) a heap-based buffer overflow in a DNS server response to the client, (2) a DNS server response with malformed ATMA records, and (3) a length miscalculation in TXT, HINFO, X25, and ISDN records."

"CVE-2005-2315: Buffer overflow in Domain Name Relay Daemon (DNRD) before 2.19.1 allows remote attackers to execute arbitrary code via a large number

of large
flags cle

"CVE-20
code for
BIND 8.
attacker
(crash) v
overflow
nameserv

"CVE-20
TFTP cl
remote r
arbitrary
that is h
function.

"CVE-20
overflow
Windows
-e option

attackers to execute
long zone name
constants represented
."

tack-based buffer
for dproxy 0.1
emote attackers to
de via a long DNS
P port 53."

tack-based buffer
shake function in
emote attackers to
de via a crafted DNS

uffer overflow in
3.1.3 and earlier
attackers to execute
malformed TCP
vents Recursor from

properly calculating the TCP DNS query length."

"CVE-2006-3441: Buffer overflow in the DNS Client service in Microsoft Windows 2000 SP4, XP SP1 and SP2, and Server 2003 SP1 allows remote attackers to execute arbitrary code via a crafted record response. NOTE: while MS06-041 implies that there is a single issue, there are multiple vectors, and likely multiple vulnerabilities, related to (1) a heap-based buffer overflow in a DNS server response to the client, (2) a DNS server response with malformed ATMA records, and (3) a length miscalculation in TXT, HINFO, X25, and ISDN records."

"CVE-2005-2315: Buffer overflow in Domain Name Relay Daemon (DNRD) before 2.19.1 allows remote attackers to execute arbitrary code via a large number

of large DNS packet
flags cleared."

"CVE-2005-0033 Bu
code for recursion a
BIND 8.4.4 and 8.4.
attackers to cause a
(crash) via queries t
overflow in the q_use
nameservers and add

"CVE-2004-1485: B
TFTP client in InetU
remote malicious DN
arbitrary code via a
that is handled by th
function."

"CVE-2004-1317: St
overflow in doexec.c
Windows 1.1, when
-e option, allows rem

execute
ame
epresented

uffer
.1
kers to
g DNS

uffer
on in
kers to
ted DNS

w in
arlier
execute
TCP
or from

properly calculating the TCP DNS query length."

"CVE-2006-3441: Buffer overflow in the DNS Client service in Microsoft Windows 2000 SP4, XP SP1 and SP2, and Server 2003 SP1 allows remote attackers to execute arbitrary code via a crafted record response. NOTE: while MS06-041 implies that there is a single issue, there are multiple vectors, and likely multiple vulnerabilities, related to (1) a heap-based buffer overflow in a DNS server response to the client, (2) a DNS server response with malformed ATMA records, and (3) a length miscalculation in TXT, HINFO, X25, and ISDN records."

"CVE-2005-2315: Buffer overflow in Domain Name Relay Daemon (DNRD) before 2.19.1 allows remote attackers to execute arbitrary code via a large number

of large DNS packets with the Z
flags cleared."

"CVE-2005-0033 Buffer overflow
code for recursion and glue fetch
BIND 8.4.4 and 8.4.5 allows rem
attackers to cause a denial of se
(crash) via queries that trigger t
overflow in the q_usedns array th
nameservers and addresses."

"CVE-2004-1485: Buffer overflow
TFTP client in InetUtils 1.4.2 al
remote malicious DNS servers to
arbitrary code via a large DNS r
that is handled by the gethostby
function."

"CVE-2004-1317: Stack-based b
overflow in doexec.c in Netcat fo
Windows 1.1, when running with
-e option, allows remote attacker

properly calculating the TCP DNS query length."

"CVE-2006-3441: Buffer overflow in the DNS Client service in Microsoft Windows 2000 SP4, XP SP1 and SP2, and Server 2003 SP1 allows remote attackers to execute arbitrary code via a crafted record response. NOTE: while MS06-041 implies that there is a single issue, there are multiple vectors, and likely multiple vulnerabilities, related to (1) a heap-based buffer overflow in a DNS server response to the client, (2) a DNS server response with malformed ATMA records, and (3) a length miscalculation in TXT, HINFO, X25, and ISDN records."

"CVE-2005-2315: Buffer overflow in Domain Name Relay Daemon (DNRD) before 2.19.1 allows remote attackers to execute arbitrary code via a large number of large DNS packets with the Z and QR flags cleared."

"CVE-2005-0033 Buffer overflow in the code for recursion and glue fetching in BIND 8.4.4 and 8.4.5 allows remote attackers to cause a denial of service (crash) via queries that trigger the overflow in the q_usedns array that tracks nameservers and addresses."

"CVE-2004-1485: Buffer overflow in the TFTP client in InetUtils 1.4.2 allows remote malicious DNS servers to execute arbitrary code via a large DNS response that is handled by the gethostbyname function."

"CVE-2004-1317: Stack-based buffer overflow in doexec.c in Netcat for Windows 1.1, when running with the -e option, allows remote attackers to

calculating the TCP DNS query

006-3441: Buffer overflow in the ent service in Microsoft Windows 4, XP SP1 and SP2, and Server 1 allows remote attackers to arbitrary code via a crafted esponse. NOTE: while MS06-041 hat there is a single issue, there iple vectors, and likely multiple ilities, related to (1) a heap- uffer overflow in a DNS server to the client, (2) a DNS server with malformed ATMA records, a length miscalculation in TXT, X25, and ISDN records."

005-2315: Buffer overflow in Name Relay Daemon (DNRD) .19.1 allows remote attackers to arbitrary code via a large number

of large DNS packets with the Z and QR flags cleared."

"CVE-2005-0033 Buffer overflow in the code for recursion and glue fetching in BIND 8.4.4 and 8.4.5 allows remote attackers to cause a denial of service (crash) via queries that trigger the overflow in the q_usedns array that tracks nameservers and addresses."

"CVE-2004-1485: Buffer overflow in the TFTP client in InetUtils 1.4.2 allows remote malicious DNS servers to execute arbitrary code via a large DNS response that is handled by the gethostbyname function."

"CVE-2004-1317: Stack-based buffer overflow in doexec.c in Netcat for Windows 1.1, when running with the -e option, allows remote attackers to

execute
comman

"CVE-20
mysql_re
4.x befo
allows re
denial of
arbitrary
large ad

"CVE-20
getaddri
2.2.2, wh
allows re
arbitrary
is obtain

"CVE-20
the rever
Daemon

the TCP DNS query

uffer overflow in the
n Microsoft Windows
and SP2, and Server
note attackers to
de via a crafted
TE: while MS06-041
a single issue, there
and likely multiple
d to (1) a heap-
w in a DNS server
t, (2) a DNS server
rmed ATMA records,
scalculation in TXT,
DN records."

uffer overflow in
Daemon (DNRD)
remote attackers to
de via a large number

of large DNS packets with the Z and QR flags cleared."

"CVE-2005-0033 Buffer overflow in the code for recursion and glue fetching in BIND 8.4.4 and 8.4.5 allows remote attackers to cause a denial of service (crash) via queries that trigger the overflow in the q_usedns array that tracks nameservers and addresses."

"CVE-2004-1485: Buffer overflow in the TFTP client in InetUtils 1.4.2 allows remote malicious DNS servers to execute arbitrary code via a large DNS response that is handled by the gethostbyname function."

"CVE-2004-1317: Stack-based buffer overflow in doexec.c in Netcat for Windows 1.1, when running with the -e option, allows remote attackers to

execute arbitrary cod
command."

"CVE-2004-0836: Bu
mysql_real_connect fu
4.x before 4.0.21, an
allows remote DNS s
denial of service and
arbitrary code via a
large address length

"CVE-2004-0150: Bu
getaddrinfo function
2.2.2, when IPv6 sup
allows remote attack
arbitrary code via an
is obtained using DN

"CVE-2003-1377: Bu
the reverse DNS loo
Daemon (SIRCD) 0.

NS query

w in the
Windows
d Server
rs to
ted
MS06-041
ue, there
nultiple
eap-
server
S server
records,
in TXT,
"

w in
DNRD)
ckers to
e number

of large DNS packets with the Z and QR flags cleared."

"CVE-2005-0033 Buffer overflow in the code for recursion and glue fetching in BIND 8.4.4 and 8.4.5 allows remote attackers to cause a denial of service (crash) via queries that trigger the overflow in the q_usedns array that tracks nameservers and addresses."

"CVE-2004-1485: Buffer overflow in the TFTP client in InetUtils 1.4.2 allows remote malicious DNS servers to execute arbitrary code via a large DNS response that is handled by the gethostbyname function."

"CVE-2004-1317: Stack-based buffer overflow in doexec.c in Netcat for Windows 1.1, when running with the -e option, allows remote attackers to

execute arbitrary code via a long
command."

"CVE-2004-0836: Buffer overflow
mysql_real_connect function in M
4.x before 4.0.21, and 3.x before
allows remote DNS servers to ca
denial of service and possibly exe
arbitrary code via a DNS respons
large address length (h_length)."

"CVE-2004-0150: Buffer overflow
getaddrinfo function in Python 2
2.2.2, when IPv6 support is disa
allows remote attackers to execu
arbitrary code via an IPv6 addres
is obtained using DNS."

"CVE-2003-1377: Buffer overflow
the reverse DNS lookup of Smar
Daemon (SIRCD) 0.4.0 and 0.4.

of large DNS packets with the Z and QR flags cleared."

"CVE-2005-0033 Buffer overflow in the code for recursion and glue fetching in BIND 8.4.4 and 8.4.5 allows remote attackers to cause a denial of service (crash) via queries that trigger the overflow in the q_usedns array that tracks nameservers and addresses."

"CVE-2004-1485: Buffer overflow in the TFTP client in InetUtils 1.4.2 allows remote malicious DNS servers to execute arbitrary code via a large DNS response that is handled by the gethostbyname function."

"CVE-2004-1317: Stack-based buffer overflow in doexec.c in Netcat for Windows 1.1, when running with the -e option, allows remote attackers to execute arbitrary code via a long DNS command."

"CVE-2004-0836: Buffer overflow in the mysql_real_connect function in MySQL 4.x before 4.0.21, and 3.x before 3.23.49, allows remote DNS servers to cause a denial of service and possibly execute arbitrary code via a DNS response with a large address length (h_length)."

"CVE-2004-0150: Buffer overflow in the getaddrinfo function in Python 2.2 before 2.2.2, when IPv6 support is disabled, allows remote attackers to execute arbitrary code via an IPv6 address that is obtained using DNS."

"CVE-2003-1377: Buffer overflow in the reverse DNS lookup of Smart IRC Daemon (SIRCD) 0.4.0 and 0.4.4 allows

DNS packets with the Z and QR
ared."

005-0033 Buffer overflow in the
recursion and glue fetching in
4.4 and 8.4.5 allows remote
s to cause a denial of service
via queries that trigger the
in the q_usedns array that tracks
vers and addresses."

004-1485: Buffer overflow in the
lient in InetUtils 1.4.2 allows
malicious DNS servers to execute
code via a large DNS response
handled by the gethostbyname
"

004-1317: Stack-based buffer
in doexec.c in Netcat for
s 1.1, when running with the
n, allows remote attackers to

execute arbitrary code via a long DNS command."

"CVE-2004-0836: Buffer overflow in the mysql_real_connect function in MySQL 4.x before 4.0.21, and 3.x before 3.23.49, allows remote DNS servers to cause a denial of service and possibly execute arbitrary code via a DNS response with a large address length (h_length)."

"CVE-2004-0150: Buffer overflow in the getaddrinfo function in Python 2.2 before 2.2.2, when IPv6 support is disabled, allows remote attackers to execute arbitrary code via an IPv6 address that is obtained using DNS."

"CVE-2003-1377: Buffer overflow in the reverse DNS lookup of Smart IRC Daemon (SIRCD) 0.4.0 and 0.4.4 allows

remote a
code via

"CVE-20
named i
earlier, a
allows re
arbitrary
response
(RR)."

"CVE-20
netstd 3
DNS ser
a long F
utilities
tftp, (4)

"CVE-20
Sendmai
to use a
TXT rec

s with the Z and QR

ffer overflow in the
nd glue fetching in
5 allows remote
denial of service
hat trigger the
edns array that tracks
lresses."

uffer overflow in the
Utils 1.4.2 allows
NS servers to execute
large DNS response
he gethostbyname

tack-based buffer
in Netcat for
running with the
note attackers to

execute arbitrary code via a long DNS command."

"CVE-2004-0836: Buffer overflow in the mysql_real_connect function in MySQL 4.x before 4.0.21, and 3.x before 3.23.49, allows remote DNS servers to cause a denial of service and possibly execute arbitrary code via a DNS response with a large address length (h_length)."

"CVE-2004-0150: Buffer overflow in the getaddrinfo function in Python 2.2 before 2.2.2, when IPv6 support is disabled, allows remote attackers to execute arbitrary code via an IPv6 address that is obtained using DNS."

"CVE-2003-1377: Buffer overflow in the reverse DNS lookup of Smart IRC Daemon (SIRCD) 0.4.0 and 0.4.4 allows

remote attackers to
code via a client wit

"CVE-2002-1219: Bu
named in BIND 4 ve
earlier, and 8 version
allows remote attack
arbitrary code via a
response containing
(RR)."

"CVE-2002-0910: Bu
netstd 3.07-17 packa
DNS servers to exec
a long FQDN reply,
utilities (1) linux-ftp
tftp, (4) traceroute,

"CVE-2002-0906: Bu
Sendmail before 8.12
to use a custom DN
TXT records, allows

Z and QR

in the
ning in
note
rvice
he
nat tracks

w in the
lows
execute
esponse
name

uffer
or
the
rs to

execute arbitrary code via a long DNS command."

"CVE-2004-0836: Buffer overflow in the mysql_real_connect function in MySQL 4.x before 4.0.21, and 3.x before 3.23.49, allows remote DNS servers to cause a denial of service and possibly execute arbitrary code via a DNS response with a large address length (h_length)."

"CVE-2004-0150: Buffer overflow in the getaddrinfo function in Python 2.2 before 2.2.2, when IPv6 support is disabled, allows remote attackers to execute arbitrary code via an IPv6 address that is obtained using DNS."

"CVE-2003-1377: Buffer overflow in the reverse DNS lookup of Smart IRC Daemon (SIRCD) 0.4.0 and 0.4.4 allows

remote attackers to execute arbi
code via a client with a long hos

"CVE-2002-1219: Buffer overflow
named in BIND 4 versions 4.9.10
earlier, and 8 versions 8.3.3 and
allows remote attackers to execu
arbitrary code via a certain DNS
response containing SIG resource
(RR)."

"CVE-2002-0910: Buffer overflow
netstd 3.07-17 package allows re
DNS servers to execute arbitrary
a long FQDN reply, as observed
utilities (1) linux-ftpd, (2) pcnfs
tftp, (4) traceroute, or (5) from

"CVE-2002-0906: Buffer overflow
Sendmail before 8.12.5, when co
to use a custom DNS map to qu
TXT records, allows remote atta

execute arbitrary code via a long DNS command."

"CVE-2004-0836: Buffer overflow in the mysql_real_connect function in MySQL 4.x before 4.0.21, and 3.x before 3.23.49, allows remote DNS servers to cause a denial of service and possibly execute arbitrary code via a DNS response with a large address length (h_length)."

"CVE-2004-0150: Buffer overflow in the getaddrinfo function in Python 2.2 before 2.2.2, when IPv6 support is disabled, allows remote attackers to execute arbitrary code via an IPv6 address that is obtained using DNS."

"CVE-2003-1377: Buffer overflow in the reverse DNS lookup of Smart IRC Daemon (SIRCD) 0.4.0 and 0.4.4 allows remote attackers to execute arbitrary code via a client with a long hostname."

"CVE-2002-1219: Buffer overflow in named in BIND 4 versions 4.9.10 and earlier, and 8 versions 8.3.3 and earlier, allows remote attackers to execute arbitrary code via a certain DNS server response containing SIG resource records (RR)."

"CVE-2002-0910: Buffer overflows in netstd 3.07-17 package allows remote DNS servers to execute arbitrary code via a long FQDN reply, as observed in the utilities (1) linux-ftpd, (2) pcnfsd, (3) tftp, (4) traceroute, or (5) from/to."

"CVE-2002-0906: Buffer overflow in Sendmail before 8.12.5, when configured to use a custom DNS map to query TXT records, allows remote attackers

arbitrary code via a long DNS
d."

004-0836: Buffer overflow in the
al_connect function in MySQL
re 4.0.21, and 3.x before 3.23.49,
emote DNS servers to cause a
f service and possibly execute
code via a DNS response with a
dress length (h_length)."

004-0150: Buffer overflow in the
nfo function in Python 2.2 before
hen IPv6 support is disabled,
emote attackers to execute
code via an IPv6 address that
ed using DNS."

003-1377: Buffer overflow in
rse DNS lookup of Smart IRC
(SIRCD) 0.4.0 and 0.4.4 allows

remote attackers to execute arbitrary code via a client with a long hostname."

"CVE-2002-1219: Buffer overflow in named in BIND 4 versions 4.9.10 and earlier, and 8 versions 8.3.3 and earlier, allows remote attackers to execute arbitrary code via a certain DNS server response containing SIG resource records (RR)."

"CVE-2002-0910: Buffer overflows in netstd 3.07-17 package allows remote DNS servers to execute arbitrary code via a long FQDN reply, as observed in the utilities (1) linux-ftpd, (2) pcnfsd, (3) tftp, (4) traceroute, or (5) from/to."

"CVE-2002-0906: Buffer overflow in Sendmail before 8.12.5, when configured to use a custom DNS map to query TXT records, allows remote attackers

to cause
execute
DNS ser

"CVE-20
the DNS
nss_ldap-
cause a
execute

"CVE-20
Internet
Microsof
remote a
code via
with a lo
reverse D
overflow

"CVE-20
resolver
network

de via a long DNS

uffer overflow in the
unction in MySQL
d 3.x before 3.23.49,
servers to cause a
possibly execute
DNS response with a
(h_length)."

uffer overflow in the
in Python 2.2 before
pport is disabled,
kers to execute
IPv6 address that
NS."

uffer overflow in
kup of Smart IRC
4.0 and 0.4.4 allows

remote attackers to execute arbitrary code via a client with a long hostname."

"CVE-2002-1219: Buffer overflow in named in BIND 4 versions 4.9.10 and earlier, and 8 versions 8.3.3 and earlier, allows remote attackers to execute arbitrary code via a certain DNS server response containing SIG resource records (RR)."

"CVE-2002-0910: Buffer overflows in netstd 3.07-17 package allows remote DNS servers to execute arbitrary code via a long FQDN reply, as observed in the utilities (1) linux-ftpd, (2) pcnfsd, (3) tftp, (4) traceroute, or (5) from/to."

"CVE-2002-0906: Buffer overflow in Sendmail before 8.12.5, when configured to use a custom DNS map to query TXT records, allows remote attackers

to cause a denial of
execute arbitrary cod
DNS server."

"CVE-2002-0825: Bu
the DNS SRV code
nss_ldap-198 allows r
cause a denial of ser
execute arbitrary cod

"CVE-2002-0698: Bu
Internet Mail Connec
Microsoft Exchange
remote attackers to
code via an EHLO r
with a long name as
reverse DNS lookup,
overflow in IMC's he

"CVE-2002-0684: Bu
resolver functions tha
network names and a

g DNS

w in the
lySQL
e 3.23.49,
use a
ecute
se with a

w in the
2.2 before
bled,
te
ss that

w in
t IRC
4 allows

remote attackers to execute arbitrary code via a client with a long hostname."

"CVE-2002-1219: Buffer overflow in named in BIND 4 versions 4.9.10 and earlier, and 8 versions 8.3.3 and earlier, allows remote attackers to execute arbitrary code via a certain DNS server response containing SIG resource records (RR)."

"CVE-2002-0910: Buffer overflows in netstd 3.07-17 package allows remote DNS servers to execute arbitrary code via a long FQDN reply, as observed in the utilities (1) linux-ftpd, (2) pcnfsd, (3) tftp, (4) traceroute, or (5) from/to."

"CVE-2002-0906: Buffer overflow in Sendmail before 8.12.5, when configured to use a custom DNS map to query TXT records, allows remote attackers

to cause a denial of service and
execute arbitrary code via a mali
DNS server."

"CVE-2002-0825: Buffer overflow
the DNS SRV code for nss_ldap
nss_ldap-198 allows remote attac
cause a denial of service and pos
execute arbitrary code."

"CVE-2002-0698: Buffer overflow
Internet Mail Connector (IMC) f
Microsoft Exchange Server 5.5 a
remote attackers to execute arbi
code via an EHLO request from
with a long name as obtained th
reverse DNS lookup, which trigg
overflow in IMC's hello response.

"CVE-2002-0684: Buffer overflow
resolver functions that perform l
network names and addresses, as

remote attackers to execute arbitrary code via a client with a long hostname."

"CVE-2002-1219: Buffer overflow in named in BIND 4 versions 4.9.10 and earlier, and 8 versions 8.3.3 and earlier, allows remote attackers to execute arbitrary code via a certain DNS server response containing SIG resource records (RR)."

"CVE-2002-0910: Buffer overflows in netstd 3.07-17 package allows remote DNS servers to execute arbitrary code via a long FQDN reply, as observed in the utilities (1) linux-ftpd, (2) pcnfsd, (3) tftp, (4) traceroute, or (5) from/to."

"CVE-2002-0906: Buffer overflow in Sendmail before 8.12.5, when configured to use a custom DNS map to query TXT records, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a malicious DNS server."

"CVE-2002-0825: Buffer overflow in the DNS SRV code for nss_ldap before nss_ldap-198 allows remote attackers to cause a denial of service and possibly execute arbitrary code."

"CVE-2002-0698: Buffer overflow in Internet Mail Connector (IMC) for Microsoft Exchange Server 5.5 allows remote attackers to execute arbitrary code via an EHLO request from a system with a long name as obtained through a reverse DNS lookup, which triggers the overflow in IMC's hello response."

"CVE-2002-0684: Buffer overflow in DNS resolver functions that perform lookup of network names and addresses, as used

attackers to execute arbitrary
a client with a long hostname."

002-1219: Buffer overflow in
n BIND 4 versions 4.9.10 and
and 8 versions 8.3.3 and earlier,
emote attackers to execute
code via a certain DNS server
containing SIG resource records

002-0910: Buffer overflows in
.07-17 package allows remote
vers to execute arbitrary code via
QDN reply, as observed in the
(1) linux-ftpd, (2) pcnfsd, (3)
traceroute, or (5) from/to."

002-0906: Buffer overflow in
l before 8.12.5, when configured
custom DNS map to query
cords, allows remote attackers

to cause a denial of service and possibly execute arbitrary code via a malicious DNS server."

"CVE-2002-0825: Buffer overflow in the DNS SRV code for nss_ldap before nss_ldap-198 allows remote attackers to cause a denial of service and possibly execute arbitrary code."

"CVE-2002-0698: Buffer overflow in Internet Mail Connector (IMC) for Microsoft Exchange Server 5.5 allows remote attackers to execute arbitrary code via an EHLO request from a system with a long name as obtained through a reverse DNS lookup, which triggers the overflow in IMC's hello response."

"CVE-2002-0684: Buffer overflow in DNS resolver functions that perform lookup of network names and addresses, as used

in BIND
and earli
servers t
a subrou
getnetby

"CVE-20
DNS res
and libbi
allows re
cause a
execute
resolvers

"CVE-20
efingerd
to 1.61,
a denial
arbitrary
an IP ad
is obtain

execute arbitrary
h a long hostname."

uffer overflow in
ersions 4.9.10 and
ns 8.3.3 and earlier,
kers to execute
certain DNS server
SIG resource records

uffer overflows in
ge allows remote
ute arbitrary code via
as observed in the
d, (2) pcnfsd, (3)
or (5) from/to."

uffer overflow in
2.5, when configured
S map to query
remote attackers

to cause a denial of service and possibly execute arbitrary code via a malicious DNS server."

"CVE-2002-0825: Buffer overflow in the DNS SRV code for nss_ldap before nss_ldap-198 allows remote attackers to cause a denial of service and possibly execute arbitrary code."

"CVE-2002-0698: Buffer overflow in Internet Mail Connector (IMC) for Microsoft Exchange Server 5.5 allows remote attackers to execute arbitrary code via an EHLO request from a system with a long name as obtained through a reverse DNS lookup, which triggers the overflow in IMC's hello response."

"CVE-2002-0684: Buffer overflow in DNS resolver functions that perform lookup of network names and addresses, as used

in BIND 4.9.8 and p
and earlier, allows re
servers to execute ar
a subroutine used by
getnetbyname and ge

"CVE-2002-0651: Bu
DNS resolver code u
and libbind, as deriv
allows remote malici
cause a denial of ser
execute arbitrary cod
resolvers."

"CVE-2002-0423: Bu
efingerd 1.5 and earl
to 1.61, allows remo
a denial of service a
arbitrary code via a
an IP address with a
is obtained via a rev

trary
stname."

w in
0 and
earlier,
te
server
e records

ws in
mote
code via
in the
d, (3)
/to."

w in
nfigured
uery
ckers

to cause a denial of service and possibly execute arbitrary code via a malicious DNS server."

"CVE-2002-0825: Buffer overflow in the DNS SRV code for nss_ldap before nss_ldap-198 allows remote attackers to cause a denial of service and possibly execute arbitrary code."

"CVE-2002-0698: Buffer overflow in Internet Mail Connector (IMC) for Microsoft Exchange Server 5.5 allows remote attackers to execute arbitrary code via an EHLO request from a system with a long name as obtained through a reverse DNS lookup, which triggers the overflow in IMC's hello response."

"CVE-2002-0684: Buffer overflow in DNS resolver functions that perform lookup of network names and addresses, as used

in BIND 4.9.8 and ported to glib
and earlier, allows remote malici
servers to execute arbitrary code
a subroutine used by functions s
getnetbyname and getnetbyaddr."

"CVE-2002-0651: Buffer overflow
DNS resolver code used in libc,
and libbind, as derived from ISC
allows remote malicious DNS ser
cause a denial of service and pos
execute arbitrary code via the st
resolvers."

"CVE-2002-0423: Buffer overflow
efingerd 1.5 and earlier, and pos
to 1.61, allows remote attackers
a denial of service and possibly e
arbitrary code via a finger reques
an IP address with a long hostna
is obtained via a reverse DNS lo

to cause a denial of service and possibly execute arbitrary code via a malicious DNS server."

"CVE-2002-0825: Buffer overflow in the DNS SRV code for nss_ldap before nss_ldap-198 allows remote attackers to cause a denial of service and possibly execute arbitrary code."

"CVE-2002-0698: Buffer overflow in Internet Mail Connector (IMC) for Microsoft Exchange Server 5.5 allows remote attackers to execute arbitrary code via an EHLO request from a system with a long name as obtained through a reverse DNS lookup, which triggers the overflow in IMC's hello response."

"CVE-2002-0684: Buffer overflow in DNS resolver functions that perform lookup of network names and addresses, as used in BIND 4.9.8 and ported to glibc 2.2.5 and earlier, allows remote malicious DNS servers to execute arbitrary code through a subroutine used by functions such as getnetbyname and getnetbyaddr."

"CVE-2002-0651: Buffer overflow in the DNS resolver code used in libc, glibc, and libbind, as derived from ISC BIND, allows remote malicious DNS servers to cause a denial of service and possibly execute arbitrary code via the stub resolvers."

"CVE-2002-0423: Buffer overflow in efingerd 1.5 and earlier, and possibly up to 1.61, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a finger request from an IP address with a long hostname that is obtained via a reverse DNS lookup."

a denial of service and possibly
arbitrary code via a malicious
ver."

002-0825: Buffer overflow in
S SRV code for nss_ldap before
-198 allows remote attackers to
denial of service and possibly
arbitrary code."

002-0698: Buffer overflow in
Mail Connector (IMC) for
ft Exchange Server 5.5 allows
attackers to execute arbitrary
an EHLO request from a system
ong name as obtained through a
DNS lookup, which triggers the
in IMC's hello response."

002-0684: Buffer overflow in DNS
functions that perform lookup of
names and addresses, as used
in BIND 4.9.8 and ported to glibc 2.2.5 and earlier, allows remote malicious DNS servers to execute arbitrary code through a subroutine used by functions such as getnetbyname and getnetbyaddr."

"CVE-2002-0651: Buffer overflow in the DNS resolver code used in libc, glibc, and libbind, as derived from ISC BIND, allows remote malicious DNS servers to cause a denial of service and possibly execute arbitrary code via the stub resolvers."

"CVE-2002-0423: Buffer overflow in efingerd 1.5 and earlier, and possibly up to 1.61, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a finger request from an IP address with a long hostname that is obtained via a reverse DNS lookup."

"CVE-20
xtell (xte
before 2.
execute
DNS hos
reverse D
string, o
request."

"CVE-20
Webalize
use rever
attackers
connecti
from an
long hos

"CVE-20
overflow
and Squ
12, 2002
attackers

service and possibly
de via a malicious

uffer overflow in
for nss_ldap before
remote attackers to
rvice and possibly
de."

uffer overflow in
ctor (IMC) for
Server 5.5 allows
execute arbitrary
equest from a system
obtained through a
which triggers the
ello response."

uffer overflow in DNS
at perform lookup of
addresses, as used

in BIND 4.9.8 and ported to glibc 2.2.5 and earlier, allows remote malicious DNS servers to execute arbitrary code through a subroutine used by functions such as getnetbyname and getnetbyaddr."

"CVE-2002-0651: Buffer overflow in the DNS resolver code used in libc, glibc, and libbind, as derived from ISC BIND, allows remote malicious DNS servers to cause a denial of service and possibly execute arbitrary code via the stub resolvers."

"CVE-2002-0423: Buffer overflow in efingerd 1.5 and earlier, and possibly up to 1.61, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a finger request from an IP address with a long hostname that is obtained via a reverse DNS lookup."

"CVE-2002-0332: Bu
xtell (xtelld) 1.91.1 a
before 2.7, allows re
execute arbitrary cod
DNS hostname that
reverse DNS lookups
string, or (3) certain
request."

"CVE-2002-0180: Bu
Webalizer 2.01-06, w
use reverse DNS loo
attackers to execute
connecting to the m
from an IP address t
long hostname."

"CVE-2002-0163: Ho
overflow in Squid be
and Squid 2.5 and 2
12, 2002 distribution
attackers to cause a

possibly
icious

w in
before
kers to
ssibly

w in
or
llows
trary
a system
rough a
ers the
."

w in DNS
ookup of
s used

in BIND 4.9.8 and ported to glibc 2.2.5 and earlier, allows remote malicious DNS servers to execute arbitrary code through a subroutine used by functions such as getnetbyname and getnetbyaddr."

"CVE-2002-0651: Buffer overflow in the DNS resolver code used in libc, glibc, and libbind, as derived from ISC BIND, allows remote malicious DNS servers to cause a denial of service and possibly execute arbitrary code via the stub resolvers."

"CVE-2002-0423: Buffer overflow in efingerd 1.5 and earlier, and possibly up to 1.61, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a finger request from an IP address with a long hostname that is obtained via a reverse DNS lookup."

"CVE-2002-0332: Buffer overflow
xtell (xtelld) 1.91.1 and earlier, a
before 2.7, allows remote attacke
execute arbitrary code via (1) a
DNS hostname that is determine
reverse DNS lookups, (2) a long
string, or (3) certain data in the
request."

"CVE-2002-0180: Buffer overflow
Webalizer 2.01-06, when configu
use reverse DNS lookups, allows
attackers to execute arbitrary co
connecting to the monitored web
from an IP address that resolves
long hostname."

"CVE-2002-0163: Heap-based bu
overflow in Squid before 2.4 STA
and Squid 2.5 and 2.6 until Mar
12, 2002 distributions, allows re
attackers to cause a denial of se

in BIND 4.9.8 and ported to glibc 2.2.5 and earlier, allows remote malicious DNS servers to execute arbitrary code through a subroutine used by functions such as getnetbyname and getnetbyaddr."

"CVE-2002-0651: Buffer overflow in the DNS resolver code used in libc, glibc, and libbind, as derived from ISC BIND, allows remote malicious DNS servers to cause a denial of service and possibly execute arbitrary code via the stub resolvers."

"CVE-2002-0423: Buffer overflow in efingerd 1.5 and earlier, and possibly up to 1.61, allows remote attackers to cause a denial of service and possibly execute arbitrary code via a finger request from an IP address with a long hostname that is obtained via a reverse DNS lookup."

"CVE-2002-0332: Buffer overflows in xtell (xtelld) 1.91.1 and earlier, and 2.x before 2.7, allows remote attackers to execute arbitrary code via (1) a long DNS hostname that is determined using reverse DNS lookups, (2) a long AUTH string, or (3) certain data in the xtell request."

"CVE-2002-0180: Buffer overflow in Webalizer 2.01-06, when configured to use reverse DNS lookups, allows remote attackers to execute arbitrary code by connecting to the monitored web server from an IP address that resolves to a long hostname."

"CVE-2002-0163: Heap-based buffer overflow in Squid before 2.4 STABLE4, and Squid 2.5 and 2.6 until March 12, 2002 distributions, allows remote attackers to cause a denial of service,

4.9.8 and ported to glibc 2.2.5
er, allows remote malicious DNS
o execute arbitrary code through
tine used by functions such as
name and getnetbyaddr."

002-0651: Buffer overflow in the
olver code used in libc, glibc,
ind, as derived from ISC BIND,
emote malicious DNS servers to
denial of service and possibly
arbitrary code via the stub
."

002-0423: Buffer overflow in
1.5 and earlier, and possibly up
allows remote attackers to cause
of service and possibly execute
code via a finger request from
ddress with a long hostname that
ed via a reverse DNS lookup."

"CVE-2002-0332: Buffer overflows in xtell (xtelld) 1.91.1 and earlier, and 2.x before 2.7, allows remote attackers to execute arbitrary code via (1) a long DNS hostname that is determined using reverse DNS lookups, (2) a long AUTH string, or (3) certain data in the xtell request."

"CVE-2002-0180: Buffer overflow in Webalizer 2.01-06, when configured to use reverse DNS lookups, allows remote attackers to execute arbitrary code by connecting to the monitored web server from an IP address that resolves to a long hostname."

"CVE-2002-0163: Heap-based buffer overflow in Squid before 2.4 STABLE4, and Squid 2.5 and 2.6 until March 12, 2002 distributions, allows remote attackers to cause a denial of service,

and poss
compress

"CVE-20
the DNS
BIND 4.
derived l
GNU glib
execute
response
(1) getn
functions

"CVE-20
in bing a
execute
hostnam
buffer af
the geth

"CVE-20
BitchX I

pported to glibc 2.2.5
emote malicious DNS
rbitrary code through
functions such as
etnetbyaddr."

uffer overflow in the
sed in libc, glibc,
ed from ISC BIND,
ous DNS servers to
rvice and possibly
de via the stub

uffer overflow in
ier, and possibly up
te attackers to cause
nd possibly execute
finger request from
a long hostname that
erse DNS lookup."

"CVE-2002-0332: Buffer overflows in xtell (xtelld) 1.91.1 and earlier, and 2.x before 2.7, allows remote attackers to execute arbitrary code via (1) a long DNS hostname that is determined using reverse DNS lookups, (2) a long AUTH string, or (3) certain data in the xtell request."

"CVE-2002-0180: Buffer overflow in Webalizer 2.01-06, when configured to use reverse DNS lookups, allows remote attackers to execute arbitrary code by connecting to the monitored web server from an IP address that resolves to a long hostname."

"CVE-2002-0163: Heap-based buffer overflow in Squid before 2.4 STABLE4, and Squid 2.5 and 2.6 until March 12, 2002 distributions, allows remote attackers to cause a denial of service,

and possibly execute
compressed DNS res

"CVE-2002-0029: Bu
the DNS stub resolv
BIND 4.9.2 through
derived libraries such
GNU glibc, allow re
execute arbitrary cod
responses that trigge
(1) getnetbyname, o
functions."

"CVE-2001-0207: Bu
in bing allows remot
execute arbitrary co
hostname, which is c
buffer after a reverse
the gethostbyaddr fu

"CVE-2001-0050: Bu
BitchX IRC client al

bc 2.2.5
ous DNS
through
uch as
"

w in the
glibc,
BIND,
rvers to
ssibly
ub

w in
sibly up
to cause
execute
st from
ame that
okup."

"CVE-2002-0332: Buffer overflows in xtell (xtelld) 1.91.1 and earlier, and 2.x before 2.7, allows remote attackers to execute arbitrary code via (1) a long DNS hostname that is determined using reverse DNS lookups, (2) a long AUTH string, or (3) certain data in the xtell request."

"CVE-2002-0180: Buffer overflow in Webalizer 2.01-06, when configured to use reverse DNS lookups, allows remote attackers to execute arbitrary code by connecting to the monitored web server from an IP address that resolves to a long hostname."

"CVE-2002-0163: Heap-based buffer overflow in Squid before 2.4 STABLE4, and Squid 2.5 and 2.6 until March 12, 2002 distributions, allows remote attackers to cause a denial of service,

and possibly execute arbitrary co
compressed DNS responses."

"CVE-2002-0029: Buffer overflow
the DNS stub resolver library in
BIND 4.9.2 through 4.9.10, and
derived libraries such as BSD lib
GNU glibc, allow remote attacke
execute arbitrary code via DNS
responses that trigger the overflo
(1) getnetbyname, or (2) getneth
functions."

"CVE-2001-0207: Buffer overflow
in bing allows remote attackers t
execute arbitrary commands via
hostname, which is copied to a
buffer after a reverse DNS looku
the gethostbyaddr function."

"CVE-2001-0050: Buffer overflow
BitchX IRC client allows remote

"CVE-2002-0332: Buffer overflows in xtell (xtelld) 1.91.1 and earlier, and 2.x before 2.7, allows remote attackers to execute arbitrary code via (1) a long DNS hostname that is determined using reverse DNS lookups, (2) a long AUTH string, or (3) certain data in the xtell request."

"CVE-2002-0180: Buffer overflow in Webalizer 2.01-06, when configured to use reverse DNS lookups, allows remote attackers to execute arbitrary code by connecting to the monitored web server from an IP address that resolves to a long hostname."

"CVE-2002-0163: Heap-based buffer overflow in Squid before 2.4 STABLE4, and Squid 2.5 and 2.6 until March 12, 2002 distributions, allows remote attackers to cause a denial of service, and possibly execute arbitrary code, via compressed DNS responses."

"CVE-2002-0029: Buffer overflows in the DNS stub resolver library in ISC BIND 4.9.2 through 4.9.10, and other derived libraries such as BSD libc and GNU glibc, allow remote attackers to execute arbitrary code via DNS server responses that trigger the overflow in the (1) getnetbyname, or (2) getnetbyaddr functions."

"CVE-2001-0207: Buffer overflow in bing allows remote attackers to execute arbitrary commands via a long hostname, which is copied to a small buffer after a reverse DNS lookup using the gethostbyaddr function."

"CVE-2001-0050: Buffer overflow in BitchX IRC client allows remote attackers

002-0332: Buffer overflows in
elld) 1.91.1 and earlier, and 2.x
.7, allows remote attackers to
arbitrary code via (1) a long
stname that is determined using
DNS lookups, (2) a long AUTH
r (3) certain data in the xtell
"

002-0180: Buffer overflow in
er 2.01-06, when configured to
rse DNS lookups, allows remote
s to execute arbitrary code by
ng to the monitored web server
IP address that resolves to a
tname."

002-0163: Heap-based buffer
in Squid before 2.4 STABLE4,
id 2.5 and 2.6 until March
2 distributions, allows remote
s to cause a denial of service,

and possibly execute arbitrary code, via compressed DNS responses."

"CVE-2002-0029: Buffer overflows in the DNS stub resolver library in ISC BIND 4.9.2 through 4.9.10, and other derived libraries such as BSD libc and GNU glibc, allow remote attackers to execute arbitrary code via DNS server responses that trigger the overflow in the (1) getnetbyname, or (2) getnetbyaddr functions."

"CVE-2001-0207: Buffer overflow in bing allows remote attackers to execute arbitrary commands via a long hostname, which is copied to a small buffer after a reverse DNS lookup using the gethostbyaddr function."

"CVE-2001-0050: Buffer overflow in BitchX IRC client allows remote attackers

to cause
execute
address
hostnam

"CVE-20
WWW p
other ve
to execu
host or
from a r

"CVE-20
nslookup
4 allows
privileges

"CVE-20
transacti
code in
to gain

uffer overflows in
and earlier, and 2.x
mote attackers to
de via (1) a long
is determined using
s, (2) a long AUTH
data in the xtell

uffer overflow in
when configured to
kups, allows remote
arbitrary code by
onitored web server
that resolves to a

eap-based buffer
fore 2.4 STABLE4,
.6 until March
s, allows remote
denial of service,

and possibly execute arbitrary code, via compressed DNS responses."

"CVE-2002-0029: Buffer overflows in the DNS stub resolver library in ISC BIND 4.9.2 through 4.9.10, and other derived libraries such as BSD libc and GNU glibc, allow remote attackers to execute arbitrary code via DNS server responses that trigger the overflow in the (1) getnetbyname, or (2) getnetbyaddr functions."

"CVE-2001-0207: Buffer overflow in bing allows remote attackers to execute arbitrary commands via a long hostname, which is copied to a small buffer after a reverse DNS lookup using the gethostbyaddr function."

"CVE-2001-0050: Buffer overflow in BitchX IRC client allows remote attackers

to cause a denial of
execute arbitrary co
address that resolves
hostname or domain

"CVE-2001-0029: Bu
WWW proxy server
other versions) allow
to execute arbitrary
host or domain nam
from a reverse DNS

"CVE-2001-0011: Bu
nslookupComplain fu
4 allows remote atta
privileges."

"CVE-2001-0010: Bu
transaction signature
code in BIND 8 allo
to gain root privilege

ws in
and 2.x
ers to
long
ed using
AUTH
xtell

w in
red to
remote
de by
p server
to a

uffer
ABLE4,
rch
mote
rvice,

and possibly execute arbitrary code, via compressed DNS responses."

"CVE-2002-0029: Buffer overflows in the DNS stub resolver library in ISC BIND 4.9.2 through 4.9.10, and other derived libraries such as BSD libc and GNU glibc, allow remote attackers to execute arbitrary code via DNS server responses that trigger the overflow in the (1) getnetbyname, or (2) getnetbyaddr functions."

"CVE-2001-0207: Buffer overflow in bing allows remote attackers to execute arbitrary commands via a long hostname, which is copied to a small buffer after a reverse DNS lookup using the gethostbyaddr function."

"CVE-2001-0050: Buffer overflow in BitchX IRC client allows remote attackers

to cause a denial of service and
execute arbitrary commands via
address that resolves to a long D
hostname or domain name."

"CVE-2001-0029: Buffer overflow
WWW proxy server 1.4.6 (and p
other versions) allows remote att
to execute arbitrary commands v
host or domain name that is obt
from a reverse DNS lookup."

"CVE-2001-0011: Buffer overflow
nslookupComplain function in BI
4 allows remote attackers to gai
privileges."

"CVE-2001-0010: Buffer overflow
transaction signature (TSIG) han
code in BIND 8 allows remote a
to gain root privileges."

and possibly execute arbitrary code, via compressed DNS responses."

"CVE-2002-0029: Buffer overflows in the DNS stub resolver library in ISC BIND 4.9.2 through 4.9.10, and other derived libraries such as BSD libc and GNU glibc, allow remote attackers to execute arbitrary code via DNS server responses that trigger the overflow in the (1) getnetbyname, or (2) getnetbyaddr functions."

"CVE-2001-0207: Buffer overflow in bing allows remote attackers to execute arbitrary commands via a long hostname, which is copied to a small buffer after a reverse DNS lookup using the gethostbyaddr function."

"CVE-2001-0050: Buffer overflow in BitchX IRC client allows remote attackers to cause a denial of service and possibly execute arbitrary commands via an IP address that resolves to a long DNS hostname or domain name."

"CVE-2001-0029: Buffer overflow in oops WWW proxy server 1.4.6 (and possibly other versions) allows remote attackers to execute arbitrary commands via a long host or domain name that is obtained from a reverse DNS lookup."

"CVE-2001-0011: Buffer overflow in nslookupComplain function in BIND 4 allows remote attackers to gain root privileges."

"CVE-2001-0010: Buffer overflow in transaction signature (TSIG) handling code in BIND 8 allows remote attackers to gain root privileges."

sibly execute arbitrary code, via
sed DNS responses."

002-0029: Buffer overflows in
S stub resolver library in ISC
9.2 through 4.9.10, and other
ibraries such as BSD libc and
bc, allow remote attackers to
arbitrary code via DNS server
s that trigger the overflow in the
etbyname, or (2) getnetbyaddr
s."

001-0207: Buffer overflow
allows remote attackers to
arbitrary commands via a long
e, which is copied to a small
fter a reverse DNS lookup using
ostbyaddr function."

001-0050: Buffer overflow in
RC client allows remote attackers

to cause a denial of service and possibly execute arbitrary commands via an IP address that resolves to a long DNS hostname or domain name."

"CVE-2001-0029: Buffer overflow in oops WWW proxy server 1.4.6 (and possibly other versions) allows remote attackers to execute arbitrary commands via a long host or domain name that is obtained from a reverse DNS lookup."

"CVE-2001-0011: Buffer overflow in nslookupComplain function in BIND 4 allows remote attackers to gain root privileges."

"CVE-2001-0010: Buffer overflow in transaction signature (TSIG) handling code in BIND 8 allows remote attackers to gain root privileges."

"CVE-20
L0pht A
to execu
malforme

"CVE-19
1.2.26 cl
could all
a denial
comman
is not pr
passing."

"CVE-19
Tetrix Te
remote a
service a
comman
from a h

"CVE-19
BIND 8.

arbitrary code, via
ponses."

uffer overflows in
er library in ISC
4.9.10, and other
as BSD libc and
mote attackers to
de via DNS server
er the overflow in the
r (2) getnetbyaddr

uffer overflow
e attackers to
mmands via a long
copied to a small
e DNS lookup using
nction."

uffer overflow in
lows remote attackers

to cause a denial of service and possibly execute arbitrary commands via an IP address that resolves to a long DNS hostname or domain name."

"CVE-2001-0029: Buffer overflow in oops WWW proxy server 1.4.6 (and possibly other versions) allows remote attackers to execute arbitrary commands via a long host or domain name that is obtained from a reverse DNS lookup."

"CVE-2001-0011: Buffer overflow in nslookupComplain function in BIND 4 allows remote attackers to gain root privileges."

"CVE-2001-0010: Buffer overflow in transaction signature (TSIG) handling code in BIND 8 allows remote attackers to gain root privileges."

"CVE-2000-0405: Bu
L0pht AntiSniff allow
to execute arbitrary
malformed DNS resp

"CVE-1999-1321: Bu
1.2.26 client with Ke
could allow remote a
a denial of service or
commands via a long
is not properly handl
passing."

"CVE-1999-1060: Bu
Tetrix TetriNet daem
remote attackers to
service and possibly
commands by connec
from a host with a l

"CVE-1999-0833: Bu
BIND 8.2 via NXT r

ode, via

ws in
ISC
other
c and
ers to
server
ow in the
byaddr

w
to
a long
small
p using

w in
attackers

to cause a denial of service and possibly execute arbitrary commands via an IP address that resolves to a long DNS hostname or domain name."

"CVE-2001-0029: Buffer overflow in oops WWW proxy server 1.4.6 (and possibly other versions) allows remote attackers to execute arbitrary commands via a long host or domain name that is obtained from a reverse DNS lookup."

"CVE-2001-0011: Buffer overflow in nslookupComplain function in BIND 4 allows remote attackers to gain root privileges."

"CVE-2001-0010: Buffer overflow in transaction signature (TSIG) handling code in BIND 8 allows remote attackers to gain root privileges."

"CVE-2000-0405: Buffer overflow
L0pht AntiSniff allows remote at
to execute arbitrary commands v
malformed DNS response packet

"CVE-1999-1321: Buffer overflow
1.2.26 client with Kerberos V en
could allow remote attackers to
a denial of service or execute arb
commands via a long DNS hostn
is not properly handled during T
passing."

"CVE-1999-1060: Buffer overflow
Tetrix TetriNet daemon 1.13.16
remote attackers to cause a deni
service and possibly execute arbi
commands by connecting to port
from a host with a long DNS ho

"CVE-1999-0833: Buffer overflow
BIND 8.2 via NXT records."

to cause a denial of service and possibly execute arbitrary commands via an IP address that resolves to a long DNS hostname or domain name."

"CVE-2001-0029: Buffer overflow in oops WWW proxy server 1.4.6 (and possibly other versions) allows remote attackers to execute arbitrary commands via a long host or domain name that is obtained from a reverse DNS lookup."

"CVE-2001-0011: Buffer overflow in nslookupComplain function in BIND 4 allows remote attackers to gain root privileges."

"CVE-2001-0010: Buffer overflow in transaction signature (TSIG) handling code in BIND 8 allows remote attackers to gain root privileges."

"CVE-2000-0405: Buffer overflow in L0pht AntiSniff allows remote attackers to execute arbitrary commands via a malformed DNS response packet."

"CVE-1999-1321: Buffer overflow in ssh 1.2.26 client with Kerberos V enabled could allow remote attackers to cause a denial of service or execute arbitrary commands via a long DNS hostname that is not properly handled during TGT ticket passing."

"CVE-1999-1060: Buffer overflow in Tetrix TetriNet daemon 1.13.16 allows remote attackers to cause a denial of service and possibly execute arbitrary commands by connecting to port 31457 from a host with a long DNS hostname."

"CVE-1999-0833: Buffer overflow in BIND 8.2 via NXT records."

a denial of service and possibly
arbitrary commands via an IP
that resolves to a long DNS
e or domain name."

001-0029: Buffer overflow in oops
proxy server 1.4.6 (and possibly
rsions) allows remote attackers
te arbitrary commands via a long
domain name that is obtained
everse DNS lookup."

001-0011: Buffer overflow in
bComplain function in BIND
remote attackers to gain root
s."

001-0010: Buffer overflow in
on signature (TSIG) handling
BIND 8 allows remote attackers
root privileges."

"CVE-2000-0405: Buffer overflow in L0pht AntiSniff allows remote attackers to execute arbitrary commands via a malformed DNS response packet."

"CVE-1999-1321: Buffer overflow in ssh 1.2.26 client with Kerberos V enabled could allow remote attackers to cause a denial of service or execute arbitrary commands via a long DNS hostname that is not properly handled during TGT ticket passing."

"CVE-1999-1060: Buffer overflow in Tetrix TetriNet daemon 1.13.16 allows remote attackers to cause a denial of service and possibly execute arbitrary commands by connecting to port 31457 from a host with a long DNS hostname."

"CVE-1999-0833: Buffer overflow in BIND 8.2 via NXT records."

"CVE-19
in FreeB
hostnam

"CVE-19
and Sola
allows ro
host nam

"CVE-19
overflow
Releases

service and possibly
mmands via an IP
s to a long DNS
name."

uffer overflow in oops
1.4.6 (and possibly
s remote attackers
commands via a long
e that is obtained
lookup."

uffer overflow in
nction in BIND
ckers to gain root

uffer overflow in
e (TSIG) handling
ws remote attackers
es."

"CVE-2000-0405: Buffer overflow in L0pht AntiSniff allows remote attackers to execute arbitrary commands via a malformed DNS response packet."

"CVE-1999-1321: Buffer overflow in ssh 1.2.26 client with Kerberos V enabled could allow remote attackers to cause a denial of service or execute arbitrary commands via a long DNS hostname that is not properly handled during TGT ticket passing."

"CVE-1999-1060: Buffer overflow in Tetrix TetriNet daemon 1.13.16 allows remote attackers to cause a denial of service and possibly execute arbitrary commands by connecting to port 31457 from a host with a long DNS hostname."

"CVE-1999-0833: Buffer overflow in BIND 8.2 via NXT records."

"CVE-1999-0299: Bu
in FreeBSD lpd thro
hostnames."

"CVE-1999-0101: Bu
and Solaris "gethostb
allows root access th
host names."

"CVE-1999-0009: In
overflow in BIND 4.9
Releases."

possibly
an IP
DNS

w in oops
possibly
tackers
via a long
tained

w in
ND
n root

w in
ndling
ttackers

"CVE-2000-0405: Buffer overflow in L0pht AntiSniff allows remote attackers to execute arbitrary commands via a malformed DNS response packet."

"CVE-1999-1321: Buffer overflow in ssh 1.2.26 client with Kerberos V enabled could allow remote attackers to cause a denial of service or execute arbitrary commands via a long DNS hostname that is not properly handled during TGT ticket passing."

"CVE-1999-1060: Buffer overflow in Tetrix TetriNet daemon 1.13.16 allows remote attackers to cause a denial of service and possibly execute arbitrary commands by connecting to port 31457 from a host with a long DNS hostname."

"CVE-1999-0833: Buffer overflow in BIND 8.2 via NXT records."

"CVE-1999-0299: Buffer overflow in FreeBSD lpd through long DN hostnames."

"CVE-1999-0101: Buffer overflow and Solaris "gethostbyname" libr allows root access through corru host names."

"CVE-1999-0009: Inverse query overflow in BIND 4.9 and BIND Releases."

"CVE-2000-0405: Buffer overflow in L0pht AntiSniff allows remote attackers to execute arbitrary commands via a malformed DNS response packet."

"CVE-1999-1321: Buffer overflow in ssh 1.2.26 client with Kerberos V enabled could allow remote attackers to cause a denial of service or execute arbitrary commands via a long DNS hostname that is not properly handled during TGT ticket passing."

"CVE-1999-1060: Buffer overflow in Tetrix TetriNet daemon 1.13.16 allows remote attackers to cause a denial of service and possibly execute arbitrary commands by connecting to port 31457 from a host with a long DNS hostname."

"CVE-1999-0833: Buffer overflow in BIND 8.2 via NXT records."

"CVE-1999-0299: Buffer overflow in FreeBSD lpd through long DNS hostnames."

"CVE-1999-0101: Buffer overflow in AIX and Solaris "gethostbyname" library call allows root access through corrupt DNS host names."

"CVE-1999-0009: Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases."

000-0405: Buffer overflow in
ntiSniff allows remote attackers
te arbitrary commands via a
ed DNS response packet."

999-1321: Buffer overflow in ssh
ient with Kerberos V enabled
ow remote attackers to cause
of service or execute arbitrary
ds via a long DNS hostname that
roperly handled during TGT ticket
"

999-1060: Buffer overflow in
etriNet daemon 1.13.16 allows
attackers to cause a denial of
nd possibly execute arbitrary
ds by connecting to port 31457
host with a long DNS hostname."

999-0833: Buffer overflow in
2 via NXT records."

"CVE-1999-0299: Buffer overflow
in FreeBSD lpd through long DNS
hostnames."

"CVE-1999-0101: Buffer overflow in AIX
and Solaris "gethostbyname" library call
allows root access through corrupt DNS
host names."

"CVE-1999-0009: Inverse query buffer
overflow in BIND 4.9 and BIND 8
Releases."

More s

What w

Attacke
through
thanks

Or thro
thanks

But wa

uffer overflow in
ws remote attackers
commands via a
ponse packet."

uffer overflow in ssh
erberos V enabled
attackers to cause
r execute arbitrary
g DNS hostname that
led during TGT ticket

uffer overflow in
non 1.13.16 allows
cause a denial of
execute arbitrary
cting to port 31457
ong DNS hostname."

uffer overflow in
records."

"CVE-1999-0299: Buffer overflow in FreeBSD lpd through long DNS hostnames."

"CVE-1999-0101: Buffer overflow in AIX and Solaris "gethostbyname" library call allows root access through corrupt DNS host names."

"CVE-1999-0009: Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases."

More security pro

What we've learn

Attacker easily b
through packet f
thanks to bad pr

Or through buffe
thanks to bad so

But wait, there's

w in
ttackers
via a
."

w in ssh
abled
cause
bitrary
name that
GT ticket

w in
allows
ial of
trary
t 31457
ostname."

w in

"CVE-1999-0299: Buffer overflow in FreeBSD lpd through long DNS hostnames."

"CVE-1999-0101: Buffer overflow in AIX and Solaris "gethostbyname" library call allows root access through corrupt DNS host names."

"CVE-1999-0009: Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases."

More security problems

What we've learned so far:

Attacker easily breaks DNS through packet forgery, thanks to bad protocol.

Or through buffer overflow thanks to bad software.

But wait, there's more!

"CVE-1999-0299: Buffer overflow in FreeBSD lpd through long DNS hostnames."

"CVE-1999-0101: Buffer overflow in AIX and Solaris "gethostbyname" library call allows root access through corrupt DNS host names."

"CVE-1999-0009: Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases."

More security problems

What we've learned so far:

Attacker easily breaks DNS through packet forgery, thanks to bad protocol.

Or through buffer overflows, thanks to bad software.

But wait, there's more!

999-0299: Buffer overflow
SD lpd through long DNS
es."

999-0101: Buffer overflow in AIX
ris "gethostbyname" library call
oot access through corrupt DNS
mes."

999-0009: Inverse query buffer
in BIND 4.9 and BIND 8
."

More security problems

What we've learned so far:

Attacker easily breaks DNS through packet forgery, thanks to bad protocol.

Or through buffer overflows, thanks to bad software.

But wait, there's more!

"CVE-20
earlier d
value fro
which al
validatio
malforme
and ECD

This bu
allowed
DNSSE

uffer overflow
ugh long DNS

uffer overflow in AIX
byname" library call
hrough corrupt DNS

verse query buffer
9 and BIND 8

More security problems

What we've learned so far:

Attacker easily breaks DNS through packet forgery, thanks to bad protocol.

Or through buffer overflows, thanks to bad software.

But wait, there's more!

"CVE-2008-5077: O
earlier does not prop
value from the EVP_
which allows remote
validation of the cer
malformed SSL/TLS
and ECDSA keys."

This bug (annou
allowed trivial fo
DNSSEC DSA si

w
NS

w in AIX
rary call
pt DNS

buffer
8

More security problems

What we've learned so far:

Attacker easily breaks DNS through packet forgery, thanks to bad protocol.

Or through buffer overflows, thanks to bad software.

But wait, there's more!

"CVE-2008-5077: OpenSSL 0.9.8
earlier does not properly check t
value from the EVP_VerifyFinal f
which allows remote attackers to
validation of the certificate chain
malformed SSL/TLS signature fo
and ECDSA keys."

This bug (announced 2009
allowed trivial forgery of
DNSSEC DSA signatures.

More security problems

What we've learned so far:

Attacker easily breaks DNS
through packet forgery,
thanks to bad protocol.

Or through buffer overflows,
thanks to bad software.

But wait, there's more!

"CVE-2008-5077: OpenSSL 0.9.8i and earlier does not properly check the return value from the EVP_VerifyFinal function, which allows remote attackers to bypass validation of the certificate chain via a malformed SSL/TLS signature for DSA and ECDSA keys."

This bug (announced 2009.01)
allowed trivial forgery of
DNSSEC DSA signatures.

More security problems

What we've learned so far:

Attacker easily breaks DNS
through packet forgery,
thanks to bad protocol.

Or through buffer overflows,
thanks to bad software.

But wait, there's more!

"CVE-2008-5077: OpenSSL 0.9.8i and earlier does not properly check the return value from the EVP_VerifyFinal function, which allows remote attackers to bypass validation of the certificate chain via a malformed SSL/TLS signature for DSA and ECDSA keys."

This bug (announced 2009.01)
allowed trivial forgery of
DNSSEC DSA signatures.

. . . which was a big deal for
the 20 people on Earth who use
DNSSEC DSA signatures.

ecurity problems

we've learned so far:

er easily breaks DNS
h packet forgery,
to bad protocol.

ough buffer overflows,
to bad software.

it, there's more!

"CVE-2008-5077: OpenSSL 0.9.8i and earlier does not properly check the return value from the EVP_VerifyFinal function, which allows remote attackers to bypass validation of the certificate chain via a malformed SSL/TLS signature for DSA and ECDSA keys."

This bug (announced 2009.01) allowed trivial forgery of DNSSEC DSA signatures.

. . . which was a big deal for the 20 people on Earth who use DNSSEC DSA signatures.

"CVE-20
control l
9.4.1, an
not set t
query-ca
attackers
query th

Docum
cache w
usable

This bu
attacke
names

Also hu
e.g., at
cache a

oblems

ned so far:

reaks DNS

orgery,

otocol.

r overflows,

ftware.

more!

"CVE-2008-5077: OpenSSL 0.9.8i and earlier does not properly check the return value from the EVP_VerifyFinal function, which allows remote attackers to bypass validation of the certificate chain via a malformed SSL/TLS signature for DSA and ECDSA keys."

This bug (announced 2009.01) allowed trivial forgery of DNSSEC DSA signatures.

. . . which was a big deal for the 20 people on Earth who use DNSSEC DSA signatures.

"CVE-2007-2925: T
control lists (ACL) i
9.4.1, and 9.5.0a1 th
not set the allow-rec
query-cache ACLs, w
attackers to make re
query the cache."

Documentation s
cache was (by de
usable only by lo

This bug hurt co
attackers easily s
names have been

Also hurt availab
e.g., attackers ea
cache as DDoS a

s

s,

"CVE-2008-5077: OpenSSL 0.9.8i and earlier does not properly check the return value from the EVP_VerifyFinal function, which allows remote attackers to bypass validation of the certificate chain via a malformed SSL/TLS signature for DSA and ECDSA keys."

This bug (announced 2009.01) allowed trivial forgery of DNSSEC DSA signatures.

. . . which was a big deal for the 20 people on Earth who use DNSSEC DSA signatures.

"CVE-2007-2925: The default ac
control lists (ACL) in ISC BIND
9.4.1, and 9.5.0a1 through 9.5.0
not set the allow-recursion and a
query-cache ACLs, which allows
attackers to make recursive quer
query the cache."

Documentation said that cache was (by default) usable only by local networ

This bug hurt confidentiali
attackers easily see which
names have been looked u

Also hurt availability:
e.g., attackers easily use
cache as DDoS amplifier.

"CVE-2008-5077: OpenSSL 0.9.8i and earlier does not properly check the return value from the EVP_VerifyFinal function, which allows remote attackers to bypass validation of the certificate chain via a malformed SSL/TLS signature for DSA and ECDSA keys."

This bug (announced 2009.01) allowed trivial forgery of DNSSEC DSA signatures.

... which was a big deal for the 20 people on Earth who use DNSSEC DSA signatures.

"CVE-2007-2925: The default access control lists (ACL) in ISC BIND 9.4.0, 9.4.1, and 9.5.0a1 through 9.5.0a5 do not set the allow-recursion and allow-query-cache ACLs, which allows remote attackers to make recursive queries and query the cache."

Documentation said that cache was (by default) usable only by local network.

This bug hurt confidentiality: attackers easily see which names have been looked up.

Also hurt availability: e.g., attackers easily use cache as DDoS amplifier.

008-5077: OpenSSL 0.9.8i and
oes not properly check the return
om the EVP_VerifyFinal function,
lows remote attackers to bypass
n of the certificate chain via a
ed SSL/TLS signature for DSA
DSA keys."

ug (announced 2009.01)
d trivial forgery of
EC DSA signatures.

ich was a big deal for
people on Earth who use
EC DSA signatures.

"CVE-2007-2925: The default access control lists (ACL) in ISC BIND 9.4.0, 9.4.1, and 9.5.0a1 through 9.5.0a5 do not set the allow-recursion and allow-query-cache ACLs, which allows remote attackers to make recursive queries and query the cache."

Documentation said that
cache was (by default)
usable only by local network.

This bug hurt confidentiality:
attackers easily see which
names have been looked up.

Also hurt availability:
e.g., attackers easily use
cache as DDoS amplifier.

"2004 Sy
DNSD D
Vulnerab
the data
server co
the requ
exploit t
to legitim
to inapp
attacks,
attacks r

Nobod
about t

penSSL 0.9.8i and
perly check the return
LVerifyFinal function,
attackers to bypass
tificate chain via a
signature for DSA

nced 2009.01)
rgery of
gnatures.

big deal for
Earth who use
gnatures.

"CVE-2007-2925: The default access control lists (ACL) in ISC BIND 9.4.0, 9.4.1, and 9.5.0a1 through 9.5.0a5 do not set the allow-recursion and allow-query-cache ACLs, which allows remote attackers to make recursive queries and query the cache."

Documentation said that
cache was (by default)
usable only by local network.

This bug hurt confidentiality:
attackers easily see which
names have been looked up.

Also hurt availability:
e.g., attackers easily use
cache as DDoS amplifier.

"2004 Symantec Ent
DNSD DNS Cache P
Vulnerability: Dnsd
the data returned fr
server contains relate
the requested record
exploit this vulnerabi
to legitimate users b
to inappropriate host
attacks, impersonatio
attacks may be poss

Nobody had told
about the bailiwi

8i and
he return
function,
bypass
via a
or DSA

.01)

or
o use

"CVE-2007-2925: The default access control lists (ACL) in ISC BIND 9.4.0, 9.4.1, and 9.5.0a1 through 9.5.0a5 do not set the allow-recursion and allow-query-cache ACLs, which allows remote attackers to make recursive queries and query the cache."

Documentation said that cache was (by default) usable only by local network.

This bug hurt confidentiality: attackers easily see which names have been looked up.

Also hurt availability: e.g., attackers easily use cache as DDoS amplifier.

"2004 Symantec Enterprise Firew
DNSD DNS Cache Poisoning
Vulnerability: Dnsd does not ens
the data returned from a remote
server contains related informatio
the requested records. An attack
exploit this vulnerability to deny
to legitimate users by redirecting
to inappropriate hosts. Man-in-t
attacks, impersonation of sites, a
attacks may be possible."

Nobody had told Symantec
about the bailiwick fix.

"CVE-2007-2925: The default access control lists (ACL) in ISC BIND 9.4.0, 9.4.1, and 9.5.0a1 through 9.5.0a5 do not set the allow-recursion and allow-query-cache ACLs, which allows remote attackers to make recursive queries and query the cache."

Documentation said that cache was (by default) usable only by local network.

This bug hurt confidentiality: attackers easily see which names have been looked up.

Also hurt availability: e.g., attackers easily use cache as DDoS amplifier.

"2004 Symantec Enterprise Firewall DNSD DNS Cache Poisoning Vulnerability: Dnsd does not ensure that the data returned from a remote DNS server contains related information about the requested records. An attacker could exploit this vulnerability to deny service to legitimate users by redirecting traffic to inappropriate hosts. Man-in-the-middle attacks, impersonation of sites, and other attacks may be possible."

Nobody had told Symantec about the bailiwick fix.

007-2925: The default access
ists (ACL) in ISC BIND 9.4.0,
nd 9.5.0a1 through 9.5.0a5 do
the allow-recursion and allow-
che ACLs, which allows remote
s to make recursive queries and
e cache."

nentation said that
was (by default)
only by local network.

ug hurt confidentiality:
ers easily see which
have been looked up.

urt availability:
ttackers easily use
as DDoS amplifier.

"2004 Symantec Enterprise Firewall DNSD DNS Cache Poisoning Vulnerability: Dnsd does not ensure that the data returned from a remote DNS server contains related information about the requested records. An attacker could exploit this vulnerability to deny service to legitimate users by redirecting traffic to inappropriate hosts. Man-in-the-middle attacks, impersonation of sites, and other attacks may be possible."

Nobody had told Symantec about the bailiwick fix.

"CVE-20
8.3.7, an
remote a
via a ma
negative
(time-to-

Cache
`micros`
to decl
for `www`
but wo
`micros`
to decl
of `www`

he default access
n ISC BIND 9.4.0,
hrough 9.5.0a5 do
cursion and allow-
which allows remote
ecursive queries and

said that
efault)
cal network.

nfidentiality:
ee which
n looked up.

ility:
sily use
mplifier.

"2004 Symantec Enterprise Firewall DNSD DNS Cache Poisoning Vulnerability: Dnsd does not ensure that the data returned from a remote DNS server contains related information about the requested records. An attacker could exploit this vulnerability to deny service to legitimate users by redirecting traffic to inappropriate hosts. Man-in-the-middle attacks, impersonation of sites, and other attacks may be possible."

Nobody had told Symantec about the bailiwick fix.

"CVE-2003-0914: IS
8.3.7, and 8.4.x befo
remote attackers to
via a malicious name
negative responses w
(time-to-live) value."

Cache wouldn't a
`microsoft.com`
to declare an ado
for `www.google.`
but would allow
`microsoft.com`
to declare *nonex*
of `www.google.`

ccess
9.4.0,
a5 do
allow-
remote
ies and

rk.

ty:

p.

"2004 Symantec Enterprise Firewall DNSD DNS Cache Poisoning
Vulnerability: Dnsd does not ensure that the data returned from a remote DNS server contains related information about the requested records. An attacker could exploit this vulnerability to deny service to legitimate users by redirecting traffic to inappropriate hosts. Man-in-the-middle attacks, impersonation of sites, and other attacks may be possible."

Nobody had told Symantec about the bailiwick fix.

"CVE-2003-0914: ISC BIND 8.3
8.3.7, and 8.4.x before 8.4.3, allo
remote attackers to poison the c
via a malicious name server that
negative responses with a large T
(time-to-live) value."

Cache wouldn't allow `microsoft.com` servers to declare an address for `www.google.com`, but would allow `microsoft.com` servers to declare *nonexistence* of `www.google.com`.

"2004 Symantec Enterprise Firewall DNSD DNS Cache Poisoning Vulnerability: Dnsd does not ensure that the data returned from a remote DNS server contains related information about the requested records. An attacker could exploit this vulnerability to deny service to legitimate users by redirecting traffic to inappropriate hosts. Man-in-the-middle attacks, impersonation of sites, and other attacks may be possible."

Nobody had told Symantec about the bailiwick fix.

"CVE-2003-0914: ISC BIND 8.3.x before 8.3.7, and 8.4.x before 8.4.3, allows remote attackers to poison the cache via a malicious name server that returns negative responses with a large TTL (time-to-live) value."

Cache wouldn't allow `microsoft.com` servers to declare an address for `www.google.com`, but would allow `microsoft.com` servers to declare *nonexistence* of `www.google.com`.

ymantec Enterprise Firewall
ONS Cache Poisoning
ility: Dnsd does not ensure that
returned from a remote DNS
ontains related information about
ested records. An attacker could
his vulnerability to deny service
mate users by redirecting traffic
ropriate hosts. Man-in-the-middle
impersonation of sites, and other
may be possible."

y had told Symantec
the bailiwick fix.

"CVE-2003-0914: ISC BIND 8.3.x before 8.3.7, and 8.4.x before 8.4.3, allows remote attackers to poison the cache via a malicious name server that returns negative responses with a large TTL (time-to-live) value."

Cache wouldn't allow `microsoft.com` servers to declare an address for `www.google.com`, but would allow `microsoft.com` servers to declare *nonexistence* of `www.google.com`.

"CVE-20
8.2.4 an
in BIND
permissi
secret k
Signatur
to obtai
DNS up

"Perfo
= "cha
Lack o
compro

Exploit
machin
where
taken

terprise Firewall
Poisoning
does not ensure that
om a remote DNS
ed information about
s. An attacker could
ility to deny service
y redirecting traffic
ts. Man-in-the-middle
on of sites, and other
ible."

Symantec
ck fix.

"CVE-2003-0914: ISC BIND 8.3.x before 8.3.7, and 8.4.x before 8.4.3, allows remote attackers to poison the cache via a malicious name server that returns negative responses with a large TTL (time-to-live) value."

Cache wouldn't allow `microsoft.com` servers to declare an address for `www.google.com`, but would allow `microsoft.com` servers to declare *nonexistence* of `www.google.com`.

"CVE-2001-0497: dn
8.2.4 and earlier, an
in BIND 9.1.2 and e
permissions for a HM
secret key file used f
Signatures (TSIG), w
to obtain the keys a
DNS updates."

"Perform dynami
= "change all yo
Lack of confident
compromises inte

Exploitable on m
machines, and on
where another se
taken over by an

wall

sure that
e DNS
on about
ker could
service
g traffic
he-middle
and other

c

"CVE-2003-0914: ISC BIND 8.3.x before 8.3.7, and 8.4.x before 8.4.3, allows remote attackers to poison the cache via a malicious name server that returns negative responses with a large TTL (time-to-live) value."

Cache wouldn't allow `microsoft.com` servers to declare an address for `www.google.com`, but would allow `microsoft.com` servers to declare *nonexistence* of `www.google.com`.

"CVE-2001-0497: dnskeygen in
8.2.4 and earlier, and dnssec-key
in BIND 9.1.2 and earlier, set in
permissions for a HMAC-MD5 sh
secret key file used for DNS Tra
Signatures (TSIG), which allows
to obtain the keys and perform
DNS updates."

"Perform dynamic DNS up
= "change all your DNS d
Lack of confidentiality of k
compromises integrity of da

Exploitable on multiuser machines, and on machines where another server has b taken over by an attacker.

"CVE-2003-0914: ISC BIND 8.3.x before 8.3.7, and 8.4.x before 8.4.3, allows remote attackers to poison the cache via a malicious name server that returns negative responses with a large TTL (time-to-live) value."

Cache wouldn't allow `microsoft.com` servers to declare an address for `www.google.com`, but would allow `microsoft.com` servers to declare *nonexistence* of `www.google.com`.

"CVE-2001-0497: dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-MD5 shared secret key file used for DNS Transactional Signatures (TSIG), which allows attackers to obtain the keys and perform dynamic DNS updates."

"Perform dynamic DNS updates" = "change all your DNS data."

Lack of confidentiality of keys compromises integrity of data.

Exploitable on multiuser machines, and on machines where another server has been taken over by an attacker.

003-0914: ISC BIND 8.3.x before
nd 8.4.x before 8.4.3, allows
attackers to poison the cache
alicious name server that returns
responses with a large TTL
-live) value."

wouldn't allow
`soft.com` servers
are an address
`w.google.com`,
uld allow
`soft.com` servers
are *nonexistence*
`.google.com`.

"CVE-2001-0497: dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-MD5 shared secret key file used for DNS Transactional Signatures (TSIG), which allows attackers to obtain the keys and perform dynamic DNS updates."

"Perform dynamic DNS updates"
= "change all your DNS data."
Lack of confidentiality of keys compromises integrity of data.

Exploitable on multiuser machines, and on machines where another server has been taken over by an attacker.

Isn't th

Public
comput
Protect
the ave
critical
and ev

C BIND 8.3.x before
ore 8.4.3, allows
poison the cache
e server that returns
with a large TTL

allow
servers
dress
.com,

servers
istence
com.

"CVE-2001-0497: dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-MD5 shared secret key file used for DNS Transactional Signatures (TSIG), which allows attackers to obtain the keys and perform dynamic DNS updates."

"Perform dynamic DNS updates"
= "change all your DNS data."
Lack of confidentiality of keys compromises integrity of data.

Exploitable on multiuser machines, and on machines where another server has been taken over by an attacker.

Isn't this embarra

Public goal of
computer-securit
Protection of
the average hom
critical-infrastruc
and everything i

.x before
ows
cache
returns
TTL

"CVE-2001-0497: dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-MD5 shared secret key file used for DNS Transactional Signatures (TSIG), which allows attackers to obtain the keys and perform dynamic DNS updates."

"Perform dynamic DNS updates"
= "change all your DNS data."
Lack of confidentiality of keys compromises integrity of data.

Exploitable on multiuser machines, and on machines where another server has been taken over by an attacker.

Isn't this embarrassing?

Public goal of
computer-security research
Protection of
the average home compute
critical-infrastructure comp
and everything in between.

"CVE-2001-0497: dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-MD5 shared secret key file used for DNS Transactional Signatures (TSIG), which allows attackers to obtain the keys and perform dynamic DNS updates."

"Perform dynamic DNS updates"
= "change all your DNS data."

Lack of confidentiality of keys compromises integrity of data.

Exploitable on multiuser machines, and on machines where another server has been taken over by an attacker.

Isn't this embarrassing?

Public goal of
computer-security research:
Protection of
the average home computer;
critical-infrastructure computers;
and everything in between.

"CVE-2001-0497: dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-MD5 shared secret key file used for DNS Transactional Signatures (TSIG), which allows attackers to obtain the keys and perform dynamic DNS updates."

"Perform dynamic DNS updates"
= "change all your DNS data."
Lack of confidentiality of keys compromises integrity of data.

Exploitable on multiuser machines, and on machines where another server has been taken over by an attacker.

Isn't this embarrassing?

Public goal of
computer-security research:
Protection of
the average home computer;
critical-infrastructure computers;
and everything in between.

Secret goal of
computer-security research:
Lifetime employment
for computer-security researchers.

001-0497: dnskeygen in BIND
d earlier, and dnssec-keygen
9.1.2 and earlier, set insecure
ons for a HMAC-MD5 shared
ey file used for DNS Transactional
es (TSIG), which allows attackers
n the keys and perform dynamic
dates."

rm dynamic DNS updates"
ange all your DNS data."
f confidentiality of keys
omises integrity of data.

table on multiuser
nes, and on machines
another server has been
over by an attacker.

Isn't this embarrassing?

Public goal of
computer-security research:
Protection of
the average home computer;
critical-infrastructure computers;
and everything in between.

Secret goal of
computer-security research:
Lifetime employment
for computer-security researchers.

ECRYP
Europe

eSTRE
Stream

2004.1
submis
Receive
cryptog

2008.0
papers
eSTRE
4 SW c

2008.0

nskeygen in BIND
d dnssec-keygen
earlier, set insecure
MAC-MD5 shared
for DNS Transactional
which allows attackers
nd perform dynamic

ic DNS updates"
our DNS data."
tiality of keys
egrity of data.

ultiuser
n machines
erver has been
attacker.

Isn't this embarrassing?

Public goal of
computer-security research:
Protection of
the average home computer;
critical-infrastructure computers;
and everything in between.

Secret goal of
computer-security research:
Lifetime employment
for computer-security researchers.

ECRYPT is a co
European crypto

eSTREAM is the
Stream Cipher P

2004.11: eSTRE
submissions of st
Receives 34 subm
cryptographers a

2008.04: After tw
papers and sever
eSTREAM select
4 SW ciphers an

2008.09: 4 SW,

BIND
gen
secure
nared
nsactional
attackers
dynamic

dates"
ata."
eys
ata.

s
been

Isn't this embarrassing?

Public goal of
computer-security research:
Protection of
the average home computer;
critical-infrastructure computers;
and everything in between.

Secret goal of
computer-security research:
Lifetime employment
for computer-security researchers.

ECRYPT is a consortium o
European crypto researche

eSTREAM is the "ECRYP
Stream Cipher Project."

2004.11: eSTREAM calls f
submissions of stream ciph
Receives 34 submissions fr
cryptographers around the

2008.04: After two hundre
papers and several confere
eSTREAM selects portfolio
4 SW ciphers and 4 HW ci

2008.09: 4 SW, 3 HW.

Isn't this embarrassing?

Public goal of
computer-security research:
Protection of
the average home computer;
critical-infrastructure computers;
and everything in between.

Secret goal of
computer-security research:
Lifetime employment
for computer-security researchers.

ECRYPT is a consortium of
European crypto researchers.

eSTREAM is the "ECRYPT
Stream Cipher Project."

2004.11: eSTREAM calls for
submissions of stream ciphers.
Receives 34 submissions from 97
cryptographers around the world.

2008.04: After two hundred
papers and several conferences,
eSTREAM selects portfolio of
4 SW ciphers and 4 HW ciphers.

2008.09: 4 SW, 3 HW.

his embarrassing?

goal of
ter-security research:
tion of
erage home computer;
-infrastructure computers;
erything in between.

goal of
ter-security research:
e employment
nputer-security researchers.

ECRYPT is a consortium of European crypto researchers.

eSTREAM is the "ECRYPT Stream Cipher Project."

2004.11: eSTREAM calls for submissions of stream ciphers. Receives 34 submissions from 97 cryptographers around the world.

2008.04: After two hundred papers and several conferences, eSTREAM selects portfolio of 4 SW ciphers and 4 HW ciphers.

2008.09: 4 SW, 3 HW.

eSTRE
are aim
passive
devices
in sens
are exc
in com
[Keys a
believe
lower-s
such de

Obviou
will be
over th

assing?

y research:

e computer;
cture computers;
n between.

y research:
ment
urity researchers.

ECRYPT is a consortium of European crypto researchers.

eSTREAM is the "ECRYPT Stream Cipher Project."

2004.11: eSTREAM calls for submissions of stream ciphers. Receives 34 submissions from 97 cryptographers around the world.

2008.04: After two hundred papers and several conferences, eSTREAM selects portfolio of 4 SW ciphers and 4 HW ciphers.

2008.09: 4 SW, 3 HW.

eSTREAM says:
are aimed at "de
passive RFID tag
devices such as n
in sensor network
are exceptionally
in computing pot
[Keys are] 80 bits
believe to be ade
lower-security ap
such devices mig

Obviously these
will be built into
over the next 5 c

ECRYPT is a consortium of
European crypto researchers.

eSTREAM is the "ECRYPT
Stream Cipher Project."

2004.11: eSTREAM calls for
submissions of stream ciphers.
Receives 34 submissions from 97
cryptographers around the world.

2008.04: After two hundred
papers and several conferences,
eSTREAM selects portfolio of
4 SW ciphers and 4 HW ciphers.

2008.09: 4 SW, 3 HW.

eSTREAM says: The HW
are aimed at "deployment
passive RFID tags or low-c
devices such as might be u
in sensor networks. Such d
are exceptionally constraine
in computing potential ...
[Keys are] 80 bits which we
believe to be adequate for
lower-security applications
such devices might be used

Obviously these ciphers
will be built into many chip
over the next 5 or 10 years

ECRYPT is a consortium of European crypto researchers.

eSTREAM is the "ECRYPT Stream Cipher Project."

2004.11: eSTREAM calls for submissions of stream ciphers. Receives 34 submissions from 97 cryptographers around the world.

2008.04: After two hundred papers and several conferences, eSTREAM selects portfolio of 4 SW ciphers and 4 HW ciphers.

2008.09: 4 SW, 3 HW.

eSTREAM says: The HW ciphers are aimed at "deployment on passive RFID tags or low-cost devices such as might be used in sensor networks. Such devices are exceptionally constrained in computing potential ... [Keys are] 80 bits which we believe to be adequate for the lower-security applications where such devices might be used."

Obviously these ciphers will be built into many chips over the next 5 or 10 years.

PT is a consortium of
ean crypto researchers.

AM is the "ECRYPT
Cipher Project."

1: eSTREAM calls for
sions of stream ciphers.
es 34 submissions from 97
graphers around the world.

4: After two hundred
and several conferences,
AM selects portfolio of
ciphers and 4 HW ciphers.

9: 4 SW, 3 HW.

eSTREAM says: The HW ciphers are aimed at "deployment on passive RFID tags or low-cost devices such as might be used in sensor networks. Such devices are exceptionally constrained in computing potential ... [Keys are] 80 bits which we believe to be adequate for the lower-security applications where such devices might be used."

Obviously these ciphers will be built into many chips over the next 5 or 10 years.

Iain De
"Asses
of key
Buy FP
break 8
Or: $1

Cost w
Will co
of more

Same s
1024-b
160-bit

nsortium of
researchers.

e "ECRYPT
roject."

AM calls for
tream ciphers.
missions from 97
round the world.

wo hundred
al conferences,
ts portfolio of
d 4 HW ciphers.

3 HW.

eSTREAM says: The HW ciphers are aimed at "deployment on passive RFID tags or low-cost devices such as might be used in sensor networks. Such devices are exceptionally constrained in computing potential . . . [Keys are] 80 bits which we believe to be adequate for the lower-security applications where such devices might be used."

Obviously these ciphers will be built into many chips over the next 5 or 10 years.

Iain Devlin, Alan
"Assessing the se
of key length," 2
Buy FPGAs for $
break 80-bit keys
Or: $165 million

Cost will continu
Will come within
of more and mor

Same story in pu
1024-bit RSA wi
160-bit ECC will

of
rs.
T
for
ers.
om 97
world.
d
nces,
o of
phers.

eSTREAM says: The HW ciphers are aimed at "deployment on passive RFID tags or low-cost devices such as might be used in sensor networks. Such devices are exceptionally constrained in computing potential . . . [Keys are] 80 bits which we believe to be adequate for the lower-security applications where such devices might be used."

Obviously these ciphers will be built into many chips over the next 5 or 10 years.

Iain Devlin, Alan Purvis, "Assessing the security of key length," 2007: Buy FPGAs for $3 million; break 80-bit keys in 1 year. Or: $165 million; 1 week.

Cost will continue to drop. Will come within reach of more and more attacker

Same story in public-key cr
1024-bit RSA will be broke
160-bit ECC will be broken

eSTREAM says: The HW ciphers are aimed at "deployment on passive RFID tags or low-cost devices such as might be used in sensor networks. Such devices are exceptionally constrained in computing potential ... [Keys are] 80 bits which we believe to be adequate for the lower-security applications where such devices might be used."

Obviously these ciphers will be built into many chips over the next 5 or 10 years.

Iain Devlin, Alan Purvis, "Assessing the security of key length," 2007: Buy FPGAs for $3 million; break 80-bit keys in 1 year. Or: $165 million; 1 week.

Cost will continue to drop. Will come within reach of more and more attackers.

Same story in public-key crypto. 1024-bit RSA will be broken. 160-bit ECC will be broken.

EAM says: The HW ciphers
ned at "deployment on
RFID tags or low-cost
s such as might be used
or networks. Such devices
ceptionally constrained
puting potential ...
are] 80 bits which we
to be adequate for the
ecurity applications where
evices might be used."

usly these ciphers
built into many chips
e next 5 or 10 years.

Iain Devlin, Alan Purvis,
"Assessing the security
of key length," 2007:
Buy FPGAs for $3 million;
break 80-bit keys in 1 year.
Or: $165 million; 1 week.

Cost will continue to drop.
Will come within reach
of more and more attackers.

Same story in public-key crypto.
1024-bit RSA will be broken.
160-bit ECC will be broken.

So user
96-bit
And th
Contin

The HW ciphers
ployment on
gs or low-cost
might be used
ks. Such devices
constrained
tential ...
s which we
equate for the
plications where
ht be used."

ciphers
many chips
or 10 years.

Iain Devlin, Alan Purvis,
"Assessing the security
of key length," 2007:
Buy FPGAs for $3 million;
break 80-bit keys in 1 year.
Or: $165 million; 1 week.

Cost will continue to drop.
Will come within reach
of more and more attackers.

Same story in public-key crypto.
1024-bit RSA will be broken.
160-bit ECC will be broken.

So users will pay
96-bit ciphers an
And then those w
Continue for dec

ciphers
on
ost
sed
levices
ed

e
the
where
d."

ps
s.

Iain Devlin, Alan Purvis,
"Assessing the security
of key length," 2007:
Buy FPGAs for $3 million;
break 80-bit keys in 1 year.
Or: $165 million; 1 week.

Cost will continue to drop.
Will come within reach
of more and more attackers.

Same story in public-key crypto.
1024-bit RSA will be broken.
160-bit ECC will be broken.

So users will pay us for
96-bit ciphers and 192-bit
And then those will be bro
Continue for decades.

Iain Devlin, Alan Purvis,
"Assessing the security
of key length," 2007:
Buy FPGAs for $3 million;
break 80-bit keys in 1 year.
Or: $165 million; 1 week.

Cost will continue to drop.
Will come within reach
of more and more attackers.

Same story in public-key crypto.
1024-bit RSA will be broken.
160-bit ECC will be broken.

So users will pay us for
96-bit ciphers and 192-bit ECC.
And then those will be broken.
Continue for decades.

Iain Devlin, Alan Purvis,
"Assessing the security
of key length," 2007:
Buy FPGAs for $3 million;
break 80-bit keys in 1 year.
Or: $165 million; 1 week.

Cost will continue to drop.
Will come within reach
of more and more attackers.

Same story in public-key crypto.
1024-bit RSA will be broken.
160-bit ECC will be broken.

So users will pay us for
96-bit ciphers and 192-bit ECC.
And then those will be broken.
Continue for decades.

Success: Lifetime employment!

This is not a new pattern.
Consider, e.g., 56-bit DES,
or 48-bit Mifare CRYPTO1,
or MD5, or Keeloq.
All breakable by brute force,
and often by faster methods.

evlin, Alan Purvis,
sing the security
length," 2007:
PGAs for $3 million;
30-bit keys in 1 year.
65 million; 1 week.

ill continue to drop.
me within reach
e and more attackers.

story in public-key crypto.
it RSA will be broken.
t ECC will be broken.

So users will pay us for
96-bit ciphers and 192-bit ECC.
And then those will be broken.
Continue for decades.

Success: Lifetime employment!

This is not a new pattern.
Consider, e.g., 56-bit DES,
or 48-bit Mifare CRYPTO1,
or MD5, or Keeloq.
All breakable by brute force,
and often by faster methods.

Naive c
from al
"There
100% s
Crypto

Purvis,
ecurity
007:
63 million;
s in 1 year.
; 1 week.

e to drop.
reach
e attackers.

blic-key crypto.
ll be broken.
be broken.

So users will pay us for
96-bit ciphers and 192-bit ECC.
And then those will be broken.
Continue for decades.

Success: Lifetime employment!

This is not a new pattern.
Consider, e.g., 56-bit DES,
or 48-bit Mifare CRYPTO1,
or MD5, or Keeloq.
All breakable by brute force,
and often by faster methods.

Naive conclusion
from all these att
"There is no such
100% secure cryp
Crypto breaks are

s.

rypto.

en.

n.

So users will pay us for
96-bit ciphers and 192-bit ECC.
And then those will be broken.
Continue for decades.

Success: Lifetime employment!

This is not a new pattern.
Consider, e.g., 56-bit DES,
or 48-bit Mifare CRYPTO1,
or MD5, or Keeloq.
All breakable by brute force,
and often by faster methods.

Naive conclusion
from all these attacks:
"There is no such thing as
100% secure cryptography!
Crypto breaks are inevitabl

So users will pay us for
96-bit ciphers and 192-bit ECC.
And then those will be broken.
Continue for decades.

Success: Lifetime employment!

This is not a new pattern.
Consider, e.g., 56-bit DES,
or 48-bit Mifare CRYPTO1,
or MD5, or Keeloq.
All breakable by brute force,
and often by faster methods.

Naive conclusion
from all these attacks:
"There is no such thing as
100% secure cryptography!
Crypto breaks are inevitable!"

So users will pay us for
96-bit ciphers and 192-bit ECC.
And then those will be broken.
Continue for decades.

Success: Lifetime employment!

This is not a new pattern.
Consider, e.g., 56-bit DES,
or 48-bit Mifare CRYPTO1,
or MD5, or Keeloq.
All breakable by brute force,
and often by faster methods.

Naive conclusion
from all these attacks:
"There is no such thing as
100% secure cryptography!
Crypto breaks are inevitable!"

This conclusion is unjustified
and almost certainly wrong.

Nobody has found patterns
in output of 256-bit AES.
Most cryptographers think
that nobody ever will.

rs will pay us for
ciphers and 192-bit ECC.
en those will be broken.
ue for decades.

s: Lifetime employment!

not a new pattern.
er, e.g., 56-bit DES,
bit Mifare CRYPTO1,
5, or Keeloq.
akable by brute force,
ten by faster methods.

Naive conclusion
from all these attacks:
"There is no such thing as
100% secure cryptography!
Crypto breaks are inevitable!"

This conclusion is unjustified
and almost certainly wrong.

Nobody has found patterns
in output of 256-bit AES.
Most cryptographers think
that nobody ever will.

"AES s
to cach

True, b
to elim
by (for

us for
d 192-bit ECC.
will be broken.
ades.

e employment!

v pattern.
6-bit DES,
CRYPTO1,
oq.
brute force,
ter methods.

Naive conclusion
from all these attacks:
"There is no such thing as
100% secure cryptography!
Crypto breaks are inevitable!"

This conclusion is unjustified
and almost certainly wrong.

Nobody has found patterns
in output of 256-bit AES.
Most cryptographers think
that nobody ever will.

"AES software le
to cache-timing a

True, but we kno
to eliminate this
by (for example)

ECC.
ken.

nent!

1,

e,
ds.

Naive conclusion
from all these attacks:
"There is no such thing as
100% secure cryptography!
Crypto breaks are inevitable!"

This conclusion is unjustified
and almost certainly wrong.

Nobody has found patterns
in output of 256-bit AES.
Most cryptographers think
that nobody ever will.

"AES software leaks keys
to cache-timing attacks!"

True, but we know how
to eliminate this problem
by (for example) bitslicing.

Naive conclusion
from all these attacks:
"There is no such thing as
100% secure cryptography!
Crypto breaks are inevitable!"

This conclusion is unjustified
and almost certainly wrong.

Nobody has found patterns
in output of 256-bit AES.
Most cryptographers think
that nobody ever will.

"AES software leaks keys
to cache-timing attacks!"

True, but we know how
to eliminate this problem
by (for example) bitslicing.

Naive conclusion
from all these attacks:
"There is no such thing as
100% secure cryptography!
Crypto breaks are inevitable!"

This conclusion is unjustified
and almost certainly wrong.

Nobody has found patterns
in output of 256-bit AES.
Most cryptographers think
that nobody ever will.

"AES software leaks keys
to cache-timing attacks!"

True, but we know how
to eliminate this problem
by (for example) bitslicing.

"Maybe secret-key crypto is okay,
but large quantum computers will
kill public-key cryptography!"

If large quantum computers are
built, they'll break RSA and ECC,
but we have replacements.
See PQCrypto workshops.

conclusion
ll these attacks:
e is no such thing as
secure cryptography!
breaks are inevitable!"

onclusion is unjustified
most certainly wrong.

y has found patterns
out of 256-bit AES.
ryptographers think
obody ever will.

"AES software leaks keys
to cache-timing attacks!"

True, but we know how
to eliminate this problem
by (for example) bitslicing.

"Maybe secret-key crypto is okay,
but large quantum computers will
kill public-key cryptography!"

If large quantum computers are
built, they'll break RSA and ECC,
but we have replacements.
See PQCrypto workshops.

Enough
How ab
of com

Flood
even m

Conver
We'll n

Viega a
there is
security

Schneie
(and th
are ine

tacks:
h thing as
ptography!
e inevitable!"

s unjustified
inly wrong.

nd patterns
-bit AES.
hers think
r will.

"AES software leaks keys
to cache-timing attacks!"

True, but we know how
to eliminate this problem
by (for example) bitslicing.

"Maybe secret-key crypto is okay,
but large quantum computers will
kill public-key cryptography!"

If large quantum computers are
built, they'll break RSA and ECC,
but we have replacements.
See PQCrypto workshops.

Enough crypto fo
How about all th
of computer secu

Flood of successf
even more than i

Conventional wis
We'll never stop

Viega and McGra
there is no such
security, attacks

Schneier: "Softw
(and therefore se
are inevitable."

!

e!"

ed

g.

s

"AES software leaks keys
to cache-timing attacks!"

True, but we know how
to eliminate this problem
by (for example) bitslicing.

"Maybe secret-key crypto is okay,
but large quantum computers will
kill public-key cryptography!"

If large quantum computers are
built, they'll break RSA and ECC,
but we have replacements.
See PQCrypto workshops.

Enough crypto for this talk.
How about all the rest
of computer security?

Flood of successful attacks
even more than in crypto.

Conventional wisdom:
We'll never stop the flood.

Viega and McGraw: "Beca
there is no such thing as 1
security, attacks will happe

Schneier: "Software bugs
(and therefore security flaw
are inevitable."

"AES software leaks keys
to cache-timing attacks!"

True, but we know how
to eliminate this problem
by (for example) bitslicing.

"Maybe secret-key crypto is okay,
but large quantum computers will
kill public-key cryptography!"

If large quantum computers are
built, they'll break RSA and ECC,
but we have replacements.
See PQCrypto workshops.

Enough crypto for this talk.
How about all the rest
of computer security?

Flood of successful attacks,
even more than in crypto.

Conventional wisdom:
We'll never stop the flood.

Viega and McGraw: "Because
there is no such thing as 100%
security, attacks will happen."

Schneier: "Software bugs
(and therefore security flaws)
are inevitable."

software leaks keys
ne-timing attacks!”

but we know how
inate this problem
example) bitslicing.

e secret-key crypto is okay,
ge quantum computers will
blic-key cryptography!”

quantum computers are
hey’ll break RSA and ECC,
have replacements.
QCrypto workshops.

Enough crypto for this talk.
How about all the rest
of computer security?

Flood of successful attacks,
even more than in crypto.

Conventional wisdom:
We’ll never stop the flood.

Viega and McGraw: “Because
there is no such thing as 100%
security, attacks will happen.”

Schneier: “Software bugs
(and therefore security flaws)
are inevitable.”

Analog
“We’ll
from E

Why n
Or: “M
but it’s

Engine
How ex
How bi
How ca

Eventu
from E

eaks keys
attacks!"

ow how
problem
bitslicing.

ey crypto is okay,
m computers will
yptography!"

computers are
k RSA and ECC,
acements.
orkshops.

Enough crypto for this talk.
How about all the rest
of computer security?

Flood of successful attacks,
even more than in crypto.

Conventional wisdom:
We'll never stop the flood.

Viega and McGraw: "Because
there is no such thing as 100%
security, attacks will happen."

Schneier: "Software bugs
(and therefore security flaws)
are inevitable."

Analogy:
"We'll never buil
from England to

Why not? "It's i
Or: "Maybe it's
but it's much too

Engineer's reacti
How expensive is
How big a tunne
How can we redu

Eventually a tunn
from England to

is okay,
ters will
y!"

s are
d ECC,

Enough crypto for this talk.
How about all the rest
of computer security?

Flood of successful attacks,
even more than in crypto.

Conventional wisdom:
We'll never stop the flood.

Viega and McGraw: "Because
there is no such thing as 100%
security, attacks will happen."

Schneier: "Software bugs
(and therefore security flaws)
are inevitable."

Analogy:
"We'll never build a tunnel
from England to France."

Why not? "It's impossible."
Or: "Maybe it's possible,
but it's much too expensiv

Engineer's reaction:
How expensive is it?
How big a tunnel *can* we b
How can we reduce the cos

Eventually a tunnel *was* bu
from England to France.

Enough crypto for this talk.
How about all the rest
of computer security?

Flood of successful attacks,
even more than in crypto.

Conventional wisdom:
We'll never stop the flood.

Viega and McGraw: "Because there is no such thing as 100% security, attacks will happen."

Schneier: "Software bugs (and therefore security flaws) are inevitable."

Analogy:
"We'll never build a tunnel from England to France."

Why not? "It's impossible."
Or: "Maybe it's possible,
but it's much too expensive."

Engineer's reaction:
How expensive is it?
How big a tunnel *can* we build?
How can we reduce the costs?

Eventually a tunnel *was* built from England to France.

n crypto for this talk.
bout all the rest
puter security?

of successful attacks,
ore than in crypto.

ntional wisdom:
ever stop the flood.

and McGraw: "Because
s no such thing as 100%
y, attacks will happen."

er: "Software bugs
herefore security flaws)
vitable."

Analogy:
"We'll never build a tunnel
from England to France."

Why not? "It's impossible."
Or: "Maybe it's possible,
but it's much too expensive."

Engineer's reaction:
How expensive is it?
How big a tunnel *can* we build?
How can we reduce the costs?

Eventually a tunnel *was* built
from England to France.

Here's

Invulne
can an
and wi

Most "
doesn't
doesn't
and wi
we hav

or this talk.
e rest
urity?

ful attacks,
n crypto.

dom:
the flood.

aw: "Because
thing as 100%
will happen."

vare bugs
ecurity flaws)

Analogy:
"We'll never build a tunnel
from England to France."

Why not? "It's impossible."
Or: "Maybe it's possible,
but it's much too expensive."

Engineer's reaction:
How expensive is it?
How big a tunnel *can* we build?
How can we reduce the costs?

Eventually a tunnel *was* built
from England to France.

Here's what I thi

Invulnerable softw
can and *will* be b
and will become

Most "security"
doesn't aim for i
doesn't contribut
and will be disca
we have invulner

k.

s,

nuse
00%
en."

ws)

Analogy:
"We'll never build a tunnel
from England to France."

Why not? "It's impossible."
Or: "Maybe it's possible,
but it's much too expensive."

Engineer's reaction:
How expensive is it?
How big a tunnel *can* we build?
How can we reduce the costs?

Eventually a tunnel *was* built
from England to France.

Here's what I think:

Invulnerable software syste
can and *will* be built,
and will become standard.

Most "security" research t
doesn't aim for invulnerabi
doesn't contribute to it,
and will be discarded once
we have invulnerable softw

Analogy:
"We'll never build a tunnel
from England to France."

Why not? "It's impossible."
Or: "Maybe it's possible,
but it's much too expensive."

Engineer's reaction:
How expensive is it?
How big a tunnel *can* we build?
How can we reduce the costs?

Eventually a tunnel *was* built
from England to France.

Here's what I think:

Invulnerable software systems
can and *will* be built,
and will become standard.

Most "security" research today
doesn't aim for invulnerability,
doesn't contribute to it,
and will be discarded once
we have invulnerable software.

y:
never build a tunnel
ngland to France."

ot? "It's impossible."
Maybe it's possible,
s much too expensive."

er's reaction:
xpensive is it?
ig a tunnel *can* we build?
an we reduce the costs?

ally a tunnel *was* built
ngland to France.

Here's what I think:

Invulnerable software systems
can and *will* be built,
and will become standard.

Most "security" research today
doesn't aim for invulnerability,
doesn't contribute to it,
and will be discarded once
we have invulnerable software.

Elimina

Bug: S
that vi
require

Securi
that vi
security

d a tunnel
France."

mpossible."
possible,
o expensive."

on:
s it?
l *can* we build?
uce the costs?

nel *was* built
France.

Here's what I think:

Invulnerable software systems
can and *will* be built,
and will become standard.

Most "security" research today
doesn't aim for invulnerability,
doesn't contribute to it,
and will be discarded once
we have invulnerable software.

Eliminating bugs

Bug: Software fe
that violates the
requirements.

Security hole: S
that violates the
security requirem

."

e."

uild?

sts?

uilt

Here's what I think:

Invulnerable software systems
can and *will* be built,
and will become standard.

Most "security" research today
doesn't aim for invulnerability,
doesn't contribute to it,
and will be discarded once
we have invulnerable software.

Eliminating bugs

Bug: Software feature
that violates the user's
requirements.

Security hole: Software fe
that violates the user's
security requirements.

Here's what I think:

Invulnerable software systems
can and *will* be built,
and will become standard.

Most "security" research today
doesn't aim for invulnerability,
doesn't contribute to it,
and will be discarded once
we have invulnerable software.

Eliminating bugs

Bug: Software feature
that violates the user's
requirements.

Security hole: Software feature
that violates the user's
security requirements.

Here's what I think:

Invulnerable software systems
can and *will* be built,
and will become standard.

Most "security" research today
doesn't aim for invulnerability,
doesn't contribute to it,
and will be discarded once
we have invulnerable software.

Eliminating bugs

Bug: Software feature
that violates the user's
requirements.

Security hole: Software feature
that violates the user's
security requirements.

Every security hole is a bug.

Is every bug a security hole?
No. Many user requirements
are not security requirements.

what I think:

erable software systems
d *will* be built,
ll become standard.

"security" research today
t aim for invulnerability,
t contribute to it,
ll be discarded once
e invulnerable software.

Eliminating bugs

Bug: Software feature
that violates the user's
requirements.

Security hole: Software feature
that violates the user's
security requirements.

Every security hole is a bug.

Is every bug a security hole?
No. Many user requirements
are not security requirements.

Everyo
we *can*
(and th
all secu
in *extre*

What a

Space-s
"The la
program
long—
The las
softwar

nk:

ware systems
built,
standard.

research today
nvulnerability,
te to it,
rded once
able software.

Eliminating bugs

Bug: Software feature
that violates the user's
requirements.

Security hole: Software feature
that violates the user's
security requirements.

Every security hole is a bug.

Is every bug a security hole?
No. Many user requirements
are not security requirements.

Everyone agrees
we *can* eliminate
(and therefore el
all security holes)
in *extremely sma*

What about large

Space-shuttle sof
"The last three v
program—each 4
long—had just o
The last 11 versi
software had a to

ms

oday
lity,

are.

Eliminating bugs

Bug: Software feature that violates the user's requirements.

Security hole: Software feature that violates the user's security requirements.

Every security hole is a bug.

Is every bug a security hole?
No. Many user requirements are not security requirements.

Everyone agrees that we *can* eliminate all bugs (and therefore eliminate all security holes) in *extremely small* program

What about larger program

Space-shuttle software:
"The last three versions of program—each 420000 line long—had just one error ea The last 11 versions of this software had a total of 17

Eliminating bugs

Bug: Software feature
that violates the user's
requirements.

Security hole: Software feature
that violates the user's
security requirements.

Every security hole is a bug.

Is every bug a security hole?
No. Many user requirements
are not security requirements.

Everyone agrees that
we *can* eliminate all bugs
(and therefore eliminate
all security holes)
in *extremely small* programs.

What about larger programs?

Space-shuttle software:
"The last three versions of the
program—each 420000 lines
long—had just one error each.
The last 11 versions of this
software had a total of 17 errors."

ating bugs

Software feature
olates the user's
ments.

ty hole: Software feature
olates the user's
y requirements.

security hole is a bug.

y bug a security hole?
any user requirements
t security requirements.

Everyone agrees that
we *can* eliminate all bugs
(and therefore eliminate
all security holes)
in *extremely small* programs.

What about larger programs?

Space-shuttle software:
"The last three versions of the
program—each 420000 lines
long—had just one error each.
The last 11 versions of this
software had a total of 17 errors."

Estima
softwar
by care
(Estima
"all bu

Meta-e
that ha
Note:

Well-kn
Drastic
of typic
by add

eature
user's

Software feature
user's
ents.

ble is a bug.

curity hole?
equirements
requirements.

Everyone agrees that
we *can* eliminate all bugs
(and therefore eliminate
all security holes)
in *extremely small* programs.

What about larger programs?

Space-shuttle software:
"The last three versions of the
program—each 420000 lines
long—had just one error each.
The last 11 versions of this
software had a total of 17 errors."

Estimate bug rat
software-enginee
by carefully revie
(Estimate is relia
"all bugs are sha

Meta-engineer pr
that have lower b
Note: progress is

Well-known exam
Drastically reduc
of typical enginee
by adding covera

eature

g.

e?

nts

nts.

Everyone agrees that
we *can* eliminate all bugs
(and therefore eliminate
all security holes)
in *extremely small* programs.

What about larger programs?

Space-shuttle software:
"The last three versions of the
program—each 420000 lines
long—had just one error each.
The last 11 versions of this
software had a total of 17 errors."

Estimate bug rate of
software-engineering proces
by carefully reviewing code
(Estimate is reliable enoug
"all bugs are shallow.")

Meta-engineer processes
that have lower bug rates.
Note: progress is *quantifie*

Well-known example:
Drastically reduce bug rate
of typical engineering proce
by adding coverage tests.

Everyone agrees that
we *can* eliminate all bugs
(and therefore eliminate
all security holes)
in *extremely small* programs.

What about larger programs?

Space-shuttle software:
"The last three versions of the
program—each 420000 lines
long—had just one error each.
The last 11 versions of this
software had a total of 17 errors."

Estimate bug rate of
software-engineering processes
by carefully reviewing code.
(Estimate is reliable enough;
"all bugs are shallow.")

Meta-engineer processes
that have lower bug rates.
Note: progress is *quantified*.

Well-known example:
Drastically reduce bug rate
of typical engineering process
by adding coverage tests.

ne agrees that
eliminate all bugs
herefore eliminate
urity holes)
emely small programs.

about larger programs?

shuttle software:
ast three versions of the
m—each 420000 lines
had just one error each.
st 11 versions of this
re had a total of 17 errors."

Estimate bug rate of
software-engineering processes
by carefully reviewing code.
(Estimate is reliable enough;
"all bugs are shallow.")

Meta-engineer processes
that have lower bug rates.
Note: progress is *quantified*.

Well-known example:
Drastically reduce bug rate
of typical engineering process
by adding coverage tests.

Examp
"Don't

Typical
copy "
quote
Inheren
simpler
but pas
Examp
format-

djbdns'
and pro
don't h
Simples

that
all bugs
iminate
)
ll programs.

er programs?

ftware:
versions of the
420000 lines
ne error each.
ons of this
otal of 17 errors."

Estimate bug rate of
software-engineering processes
by carefully reviewing code.
(Estimate is reliable enough;
"all bugs are shallow.")

Meta-engineer processes
that have lower bug rates.
Note: progress is *quantified*.

Well-known example:
Drastically reduce bug rate
of typical engineering process
by adding coverage tests.

Example where d
"Don't parse."

Typical user inte
copy "normal" in
quote "abnormal
Inherently bug-pr
simpler copying i
but passes "norm
Example (1996 B
format-string dan

djbdns's internal
and program-leve
don't have excep
Simplest code is

ns.

ns?

the
es
ach.
s
errors.”

Estimate bug rate of
software-engineering processes
by carefully reviewing code.
(Estimate is reliable enough;
“all bugs are shallow.”)

Meta-engineer processes
that have lower bug rates.
Note: progress is *quantified*.

Well-known example:
Drastically reduce bug rate
of typical engineering process
by adding coverage tests.

Example where djbdns did
“Don’t parse.”

Typical user interfaces
copy “normal” inputs and
quote “abnormal” inputs.
Inherently bug-prone:
simpler copying is wrong
but passes “normal” tests.
Example (1996 Bernstein):
format-string danger in `log`

djbdns’s internal file struct
and program-level interface
don’t have exceptional case
Simplest code is correct co

Estimate bug rate of
software-engineering processes
by carefully reviewing code.
(Estimate is reliable enough;
"all bugs are shallow.")

Meta-engineer processes
that have lower bug rates.
Note: progress is *quantified*.

Well-known example:
Drastically reduce bug rate
of typical engineering process
by adding coverage tests.

Example where djbdns did well:
"Don't parse."

Typical user interfaces
copy "normal" inputs and
quote "abnormal" inputs.
Inherently bug-prone:
simpler copying is wrong
but passes "normal" tests.
Example (1996 Bernstein):
format-string danger in `logger`.

djbdns's internal file structures
and program-level interfaces
don't have exceptional cases.
Simplest code is correct code.

te bug rate of
re-engineering processes
efully reviewing code.
ate is reliable enough;
gs are shallow.")

engineer processes
ave lower bug rates.
progress is *quantified*.

nown example:
cally reduce bug rate
cal engineering process
ing coverage tests.

Example where djbdns did well:
"Don't parse."

Typical user interfaces
copy "normal" inputs and
quote "abnormal" inputs.
Inherently bug-prone:
simpler copying is wrong
but passes "normal" tests.
Example (1996 Bernstein):
format-string danger in `logger`.

djbdns's internal file structures
and program-level interfaces
don't have exceptional cases.
Simplest code is correct code.

Examp
integer

In C et
exactly
but *oc*

To dete
need to
Extra w

To gua
extendi
failing
need to
Extra w

e of
ring processes
wing code.
ble enough;
llow.")

rocesses
bug rates.
s *quantified*.

nple:
e bug rate
ering process
ge tests.

Example where djbdns did well:
"Don't parse."

Typical user interfaces
copy "normal" inputs and
quote "abnormal" inputs.
Inherently bug-prone:
simpler copying is wrong
but passes "normal" tests.
Example (1996 Bernstein):
format-string danger in `logger`.

djbdns's internal file structures
and program-level interfaces
don't have exceptional cases.
Simplest code is correct code.

Example where d
integer arithmeti

In C et al., a + b
exactly what it s
but *occasionally*

To detect these c
need to check fo
Extra work for pr

To guarantee sa
extending intege
failing only if out
need to use large
Extra work for pr

sses
.
h;
d.
e
ess

Example where djbdns did well:
"Don't parse."

Typical user interfaces
copy "normal" inputs and
quote "abnormal" inputs.
Inherently bug-prone:
simpler copying is wrong
but passes "normal" tests.
Example (1996 Bernstein):
format-string danger in `logger`.

djbdns's internal file structures
and program-level interfaces
don't have exceptional cases.
Simplest code is correct code.

Example where djbdns did
integer arithmetic.

In C et al., `a + b` *usually* m
exactly what it says,
but *occasionally* doesn't.

To detect these occasions,
need to check for overflows
Extra work for programmer

To guarantee sane semanti
extending integer range an
failing only if out of memo
need to use large-integer li
Extra work for programmer

Example where djbdns did well:
"Don't parse."

Typical user interfaces
copy "normal" inputs and
quote "abnormal" inputs.
Inherently bug-prone:
simpler copying is wrong
but passes "normal" tests.
Example (1996 Bernstein):
format-string danger in `logger`.

djbdns's internal file structures
and program-level interfaces
don't have exceptional cases.
Simplest code is correct code.

Example where djbdns did badly:
integer arithmetic.

In C et al., `a + b` *usually* means
exactly what it says,
but *occasionally* doesn't.

To detect these occasions,
need to check for overflows.
Extra work for programmer.

To guarantee sane semantics,
extending integer range and
failing only if out of memory,
need to use large-integer library.
Extra work for programmer.

le where djbdns did well:
parse."

l user interfaces
normal" inputs and
"abnormal" inputs.
ntly bug-prone:
r copying is wrong
sses "normal" tests.
le (1996 Bernstein):
-string danger in `logger`.

's internal file structures
ogram-level interfaces
have exceptional cases.
st code is correct code.

Example where djbdns did badly:
integer arithmetic.

In C et al., `a + b` *usually* means
exactly what it says,
but *occasionally* doesn't.

To detect these occasions,
need to check for overflows.
Extra work for programmer.

To guarantee sane semantics,
extending integer range and
failing only if out of memory,
need to use large-integer library.
Extra work for programmer.

The cl
has co
(Gunin
of an u

Fortun
was lim
and th
but thi

Anti-bu
Use lar
means

djbdns did well:

rfaces
nputs and
" inputs.
rone:
s wrong
nal" tests.
Bernstein):
nger in `logger`.

file structures
el interfaces
tional cases.
correct code.

Example where djbdns did badly:
integer arithmetic.

In C et al., `a + b` *usually* means
exactly what it says,
but *occasionally* doesn't.

To detect these occasions,
need to check for overflows.
Extra work for programmer.

To guarantee sane semantics,
extending integer range and
failing only if out of memory,
need to use large-integer library.
Extra work for programmer.

The closest that
has come to a se
(Guninski): pote
of an unchecked

Fortunately, cour
was limited by m
and thus by conf
but this was pure

Anti-bug meta-er
Use language wh
means exactly wh

well:

`gger`.

ures

es

es.

ode.

Example where djbdns did badly:
integer arithmetic.

In C et al., `a + b` *usually* means
exactly what it says,
but *occasionally* doesn't.

To detect these occasions,
need to check for overflows.
Extra work for programmer.

To guarantee sane semantics,
extending integer range and
failing only if out of memory,
need to use large-integer library.
Extra work for programmer.

The closest that qmail
has come to a security hole
(Guninski): potential overf
of an unchecked counter.

Fortunately, counter growt
was limited by memory
and thus by configuration,
but this was pure luck.

Anti-bug meta-engineering
Use language where `a + b`
means exactly what it says

Example where djbdns did badly:
integer arithmetic.

In C et al., `a + b` *usually* means
exactly what it says,
but *occasionally* doesn't.

To detect these occasions,
need to check for overflows.
Extra work for programmer.

To guarantee sane semantics,
extending integer range and
failing only if out of memory,
need to use large-integer library.
Extra work for programmer.

The closest that qmail
has come to a security hole
(Guninski): potential overflow
of an unchecked counter.

Fortunately, counter growth
was limited by memory
and thus by configuration,
but this was pure luck.

Anti-bug meta-engineering:
Use language where `a + b`
means exactly what it says.

le where djbdns did badly:
arithmetic.

al., a + b *usually* means
what it says,
casionally doesn't.

ect these occasions,
o check for overflows.
work for programmer.

rantee sane semantics,
ing integer range and
only if out of memory,
o use large-integer library.
work for programmer.

The closest that qmail
has come to a security hole
(Guninski): potential overflow
of an unchecked counter.

Fortunately, counter growth
was limited by memory
and thus by configuration,
but this was pure luck.

Anti-bug meta-engineering:
Use language where a + b
means exactly what it says.

Securit
(2009.0
of an u
copied

Decom
produc
Problem
data fr

Impact
lsec.b
from a
the thi
cache e
not jus

ljbdns did badly:
c.

usually means
ays,
doesn't.

occasions,
r overflows.
rogrammer.

ne semantics,
r range and
t of memory,
e-integer library.
rogrammer.

The closest that qmail
has come to a security hole
(Guninski): potential overflow
of an unchecked counter.

Fortunately, counter growth
was limited by memory
and thus by configuration,
but this was pure luck.

Anti-bug meta-engineering:
Use language where `a + b`
means exactly what it says.

Security hole in
(2009.02.25 Dem
of an unchecked
copied into comp

Decompressing t
produces incorre
Problem for pack
data from differe

Impact: If admin
`lsec.be` copies
from an untruste
the third party ca
cache entries for
not just `foo.lse`

badly:

neans

s.

r.

cs,

d

ry,

brary.

r.

The closest that qmail
has come to a security hole
(Guninski): potential overflow
of an unchecked counter.

Fortunately, counter growth
was limited by memory
and thus by configuration,
but this was pure luck.

Anti-bug meta-engineering:
Use language where `a + b`
means exactly what it says.

Security hole in djbdns
(2009.02.25 Dempsky): ov
of an unchecked counter
copied into compressed pac

Decompressing those packe
produces incorrect results.
Problem for packets that m
data from different sources

Impact: If administrator of
`lsec.be` copies `foo.lsec`
from an untrusted third pa
the third party can control
cache entries for `lsec.be`,
not just `foo.lsec.be`.

The closest that qmail
has come to a security hole
(Guninski): potential overflow
of an unchecked counter.

Fortunately, counter growth
was limited by memory
and thus by configuration,
but this was pure luck.

Anti-bug meta-engineering:
Use language where `a + b`
means exactly what it says.

Security hole in djbdns
(2009.02.25 Dempsky): overflow
of an unchecked counter
copied into compressed packets.

Decompressing those packets
produces incorrect results.
Problem for packets that mix
data from different sources.

Impact: If administrator of
`lsec.be` copies `foo.lsec.be`
from an untrusted third party,
the third party can control
cache entries for `lsec.be`,
not just `foo.lsec.be`.

osest that qmail
me to a security hole
ski): potential overflow
unchecked counter.

ately, counter growth
nited by memory
us by configuration,
s was pure luck.

ug meta-engineering:
nguage where `a + b`
exactly what it says.

Security hole in djbdns
(2009.02.25 Dempsky): overflow
of an unchecked counter
copied into compressed packets.

Decompressing those packets
produces incorrect results.
Problem for packets that mix
data from different sources.

Impact: If administrator of
`lsec.be` copies `foo.lsec.be`
from an untrusted third party,
the third party can control
cache entries for `lsec.be`,
not just `foo.lsec.be`.

"Large-

That's
We nee
and we
even if
than ou
Tomorr

Most C
by a ve
of all t
Most la
are rem
Occasio
handle

qmail
ecurity hole
ntial overflow
counter.

nter growth
emory
iguration,
e luck.

ngineering:
ere `a + b`
hat it says.

Security hole in djbdns
(2009.02.25 Dempsky): overflow
of an unchecked counter
copied into compressed packets.

Decompressing those packets
produces incorrect results.
Problem for packets that mix
data from different sources.

Impact: If administrator of
`lsec.be` copies `foo.lsec.be`
from an untrusted third party,
the third party can control
cache entries for `lsec.be`,
not just `foo.lsec.be`.

"Large-integer lib

That's a silly obj
We need invulner
and we need ther
even if they are 1
than our current
Tomorrow we'll n

Most CPU time
by a very small p
of all the system
Most large-intege
are removed by s
Occasional excep
handled manually

e
flow

h

:

.

Security hole in djbdns
(2009.02.25 Dempsky): overflow
of an unchecked counter
copied into compressed packets.

Decompressing those packets
produces incorrect results.
Problem for packets that mix
data from different sources.

Impact: If administrator of
`lsec.be` copies `foo.lsec.be`
from an untrusted third party,
the third party can control
cache entries for `lsec.be`,
not just `foo.lsec.be`.

"Large-integer libraries are

That's a silly objection.
We need invulnerable syste
and we need them today,
even if they are 10× slowe
than our current systems.
Tomorrow we'll make them

Most CPU time is consume
by a very small portion
of all the system's code.
Most large-integer overhea
are removed by smart com
Occasional exceptions can
handled manually at low c

Security hole in djbdns
(2009.02.25 Dempsky): overflow
of an unchecked counter
copied into compressed packets.

Decompressing those packets
produces incorrect results.
Problem for packets that mix
data from different sources.

Impact: If administrator of
`lsec.be` copies `foo.lsec.be`
from an untrusted third party,
the third party can control
cache entries for `lsec.be`,
not just `foo.lsec.be`.

"Large-integer libraries are slow!"

That's a silly objection.
We need invulnerable systems,
and we need them today,
even if they are 10× slower
than our current systems.
Tomorrow we'll make them faster.

Most CPU time is consumed
by a very small portion
of all the system's code.
Most large-integer overheads
are removed by smart compilers.
Occasional exceptions can be
handled manually at low cost.

y hole in djbdns
02.25 Dempsky): overflow
unchecked counter
into compressed packets.

pressing those packets
es incorrect results.
m for packets that mix
om different sources.

: If administrator of
be copies `foo.lsec.be`
n untrusted third party,
rd party can control
entries for `lsec.be`,
t `foo.lsec.be`.

"Large-integer libraries are slow!"

That's a silly objection.
We need invulnerable systems,
and we need them today,
even if they are 10× slower
than our current systems.
Tomorrow we'll make them faster.

Most CPU time is consumed
by a very small portion
of all the system's code.
Most large-integer overheads
are removed by smart compilers.
Occasional exceptions can be
handled manually at low cost.

More a
exampl
automa
partitio
to mak
automa
"summ
abstrac

"Okay,
much s
But in
we'll st
includi

djbdns
npsky): overflow
counter
pressed packets.

hose packets
ct results.
kets that mix
nt sources.

istrator of
`foo.lsec.be`
d third party,
an control
`lsec.be`,
`ec.be`.

"Large-integer libraries are slow!"

That's a silly objection.
We need invulnerable systems,
and we need them today,
even if they are 10× slower
than our current systems.
Tomorrow we'll make them faster.

Most CPU time is consumed
by a very small portion
of all the system's code.
Most large-integer overheads
are removed by smart compilers.
Occasional exceptions can be
handled manually at low cost.

More anti-bug m
examples in my c
automatic array
partitioning varia
to make data flo
automatic updat
"summary" varia
abstraction for te

"Okay, we can a
much smaller bu
But in a large sy
we'll still have m
including many s

erflow

ckets.

ets

nix

s.

f

.be

rty,

"Large-integer libraries are slow!"

That's a silly objection.
We need invulnerable systems,
and we need them today,
even if they are 10× slower
than our current systems.
Tomorrow we'll make them faster.

Most CPU time is consumed
by a very small portion
of all the system's code.
Most large-integer overheads
are removed by smart compilers.
Occasional exceptions can be
handled manually at low cost.

More anti-bug meta-engine
examples in my qmail pape
automatic array extensions
partitioning variables
to make data flow visible;
automatic updates of
"summary" variables;
abstraction for testability.

"Okay, we can achieve
much smaller bug rates.
But in a large system
we'll still have many bugs,
including many security ho

"Large-integer libraries are slow!"

That's a silly objection.
We need invulnerable systems,
and we need them today,
even if they are 10× slower
than our current systems.
Tomorrow we'll make them faster.

Most CPU time is consumed
by a very small portion
of all the system's code.
Most large-integer overheads
are removed by smart compilers.
Occasional exceptions can be
handled manually at low cost.

More anti-bug meta-engineering
examples in my qmail paper:
automatic array extensions;
partitioning variables
to make data flow visible;
automatic updates of
"summary" variables;
abstraction for testability.

"Okay, we can achieve
much smaller bug rates.
But in a large system
we'll still have many bugs,
including many security holes!"

-integer libraries are slow!”

a silly objection.
ed invulnerable systems,
e need them today,
they are 10× slower
ur current systems.
row we’ll make them faster.

CPU time is consumed
ery small portion
he system’s code.
arge-integer overheads
noved by smart compilers.
onal exceptions can be
d manually at low cost.

More anti-bug meta-engineering
examples in my qmail paper:
automatic array extensions;
partitioning variables
to make data flow visible;
automatic updates of
“summary” variables;
abstraction for testability.

“Okay, we can achieve
much smaller bug rates.
But in a large system
we’ll still have many bugs,
including many security holes!”

Elimina

Measur
softwar

Meta-e
that sp
to get

Note:

This is
of softw
Combi
with re

braries are slow!"

ection.
rable systems,
m today,
10× slower
systems.
make them faster.

is consumed
portion
's code.
er overheads
smart compilers.
tions can be
y at low cost.

More anti-bug meta-engineering
examples in my qmail paper:
automatic array extensions;
partitioning variables
to make data flow visible;
automatic updates of
"summary" variables;
abstraction for testability.

"Okay, we can achieve
much smaller bug rates.
But in a large system
we'll still have many bugs,
including many security holes!"

Eliminating code

Measure code ra
software-enginee

Meta-engineer pr
that spend less c
to get the same
Note: progress is

This is another c
of software-engin
Combines reason
with reducing bu

slow!”

ems,

r

n faster.

ed

ds

pilers.

be

ost.

More anti-bug meta-engineering
examples in my qmail paper:
automatic array extensions;
partitioning variables
to make data flow visible;
automatic updates of
“summary” variables;
abstraction for testability.

“Okay, we can achieve
much smaller bug rates.
But in a large system
we’ll still have many bugs,
including many security holes!”

Eliminating code

Measure code rate of
software-engineering proces

Meta-engineer processes
that spend less code
to get the same job done.
Note: progress is *quantifie*

This is another classic topi
of software-engineering res
Combines reasonably well
with reducing bug rate.

More anti-bug meta-engineering
examples in my qmail paper:
automatic array extensions;
partitioning variables
to make data flow visible;
automatic updates of
"summary" variables;
abstraction for testability.

"Okay, we can achieve
much smaller bug rates.
But in a large system
we'll still have many bugs,
including many security holes!"

Eliminating code

Measure code rate of
software-engineering processes.

Meta-engineer processes
that spend less code
to get the same job done.
Note: progress is *quantified*.

This is another classic topic
of software-engineering research.
Combines reasonably well
with reducing bug rate.

nti-bug meta-engineering
es in my qmail paper:
atic array extensions;
oning variables
ke data flow visible;
atic updates of
nary" variables;
ction for testability.

we can achieve
smaller bug rates.
a large system
till have many bugs,
ng many security holes!"

Eliminating code

Measure code rate of
software-engineering processes.

Meta-engineer processes
that spend less code
to get the same job done.
Note: progress is *quantified*.

This is another classic topic
of software-engineering research.
Combines reasonably well
with reducing bug rate.

Examp
reusing

A story
My `.f`
creatin
Surpris
someti

How S
Check
(Prohib
Extract
Keep t
of instr

Many c

eta-engineering
qmail paper:
extensions;
bles
w visible;
es of
bles;
estability.

chieve
g rates.
stem
any bugs,
ecurity holes!”

Eliminating code

Measure code rate of
software-engineering processes.

Meta-engineer processes
that spend less code
to get the same job done.
Note: progress is *quantified*.

This is another classic topic
of software-engineering research.
Combines reasonably well
with reducing bug rate.

Example where q
reusing access-co

A story from twe
My `.forward` ra
creating a new fi
Surprise: the pro
sometimes run u

How Sendmail ha
Check whether u
(Prohibit symlink
Extract delivery i
Keep track (ofte
of instructions ar

Many disastrous

eering
er:
;

les!”

Eliminating code

Measure code rate of
software-engineering processes.

Meta-engineer processes
that spend less code
to get the same job done.
Note: progress is *quantified*.

This is another classic topic
of software-engineering research.
Combines reasonably well
with reducing bug rate.

Example where qmail did w
reusing access-control code

A story from twenty years
My `.forward` ran a progra
creating a new file in `/tmp`
Surprise: the program was
sometimes run under anoth

How Sendmail handles `.fo`
Check whether user can re
(Prohibit symlinks to secre
Extract delivery instruction
Keep track (often via queu
of instructions and user.

Many disastrous bugs here

Eliminating code

Measure code rate of
software-engineering processes.

Meta-engineer processes
that spend less code
to get the same job done.
Note: progress is *quantified*.

This is another classic topic
of software-engineering research.
Combines reasonably well
with reducing bug rate.

Example where qmail did well:
reusing access-control code.

A story from twenty years ago:
My `.forward` ran a program
creating a new file in `/tmp`.
Surprise: the program was
sometimes run under another uid!

How Sendmail handles `.forward`:
Check whether user can read it.
(Prohibit symlinks to secrets!)
Extract delivery instructions.
Keep track (often via queue file)
of instructions and user.

Many disastrous bugs here.

ating code

re code rate of
re-engineering processes.

engineer processes
end less code
the same job done.
progress is *quantified*.

another classic topic
ware-engineering research.
nes reasonably well
educing bug rate.

Example where qmail did well:
reusing access-control code.

A story from twenty years ago:
My `.forward` ran a program
creating a new file in `/tmp`.
Surprise: the program was
sometimes run under another uid!

How Sendmail handles `.forward`:
Check whether user can read it.
(Prohibit symlinks to secrets!)
Extract delivery instructions.
Keep track (often via queue file)
of instructions and user.

Many disastrous bugs here.

Kernel
Kernel
Why n

How q
Start q
under t

When
the use
the ker
When
progra
the sa
No ext

te of
ring processes.

rocesses
ode
job done.
s *quantified*.

lassic topic
neering research.
ably well
g rate.

Example where qmail did well:
reusing access-control code.

A story from twenty years ago:
My `.forward` ran a program
creating a new file in `/tmp`.
Surprise: the program was
sometimes run under another uid!

How Sendmail handles `.forward`:
Check whether user can read it.
(Prohibit symlinks to secrets!)
Extract delivery instructions.
Keep track (often via queue file)
of instructions and user.

Many disastrous bugs here.

Kernel already tr
Kernel already ch
Why not reuse th

How qmail delive
Start `qmail-loc`
under the right u

When `qmail-lo`
the user's deliver
the kernel checks
When `qmail-lo`
program, the ker
the same uid to t
No extra code re

sses.

d.

c

earch.

Example where qmail did well:
reusing access-control code.

A story from twenty years ago:
My `.forward` ran a program
creating a new file in `/tmp`.
Surprise: the program was
sometimes run under another uid!

How Sendmail handles `.forward`:
Check whether user can read it.
(Prohibit symlinks to secrets!)
Extract delivery instructions.
Keep track (often via queue file)
of instructions and user.

Many disastrous bugs here.

Kernel already tracks users
Kernel already checks read
Why not reuse this code?

How qmail delivers to a us
Start `qmail-local`
under the right uid.

When `qmail-local` reads
the user's delivery instructi
the kernel checks readabilit
When `qmail-local` runs a
program, the kernel assigns
the same uid to that progr
No extra code required!

Example where qmail did well:
reusing access-control code.

A story from twenty years ago:
My `.forward` ran a program
creating a new file in `/tmp`.
Surprise: the program was
sometimes run under another uid!

How Sendmail handles `.forward`:
Check whether user can read it.
(Prohibit symlinks to secrets!)
Extract delivery instructions.
Keep track (often via queue file)
of instructions and user.

Many disastrous bugs here.

Kernel already tracks users.
Kernel already checks readability.
Why not reuse this code?

How qmail delivers to a user:
Start `qmail-local`
under the right uid.

When `qmail-local` reads
the user's delivery instructions,
the kernel checks readability.
When `qmail-local` runs a
program, the kernel assigns
the same uid to that program.
No extra code required!

le where qmail did well:
g access-control code.

y from twenty years ago:
`orward` ran a program
g a new file in `/tmp`.
e: the program was
mes run under another uid!

endmail handles `.forward`:
whether user can read it.
bit symlinks to secrets!)
t delivery instructions.
rack (often via queue file)
ructions and user.

disastrous bugs here.

Kernel already tracks users.
Kernel already checks readability.
Why not reuse this code?

How qmail delivers to a user:
Start `qmail-local`
under the right uid.

When `qmail-local` reads
the user's delivery instructions,
the kernel checks readability.
When `qmail-local` runs a
program, the kernel assigns
the same uid to that program.
No extra code required!

Examp
did bad

djbdns
of conc
About
checkin

Same f
Easy to
"if `ipm`
`qmail-`
(fixed i

Easily

qmail did well:
ontrol code.

enty years ago:
n a program
le in `/tmp`.
ogram was
nder another uid!

andles `.forward`:
ser can read it.
ks to secrets!)
instructions.
n via queue file)
nd user.

bugs here.

Kernel already tracks users.
Kernel already checks readability.
Why not reuse this code?

How qmail delivers to a user:
Start `qmail-local`
under the right uid.

When `qmail-local` reads
the user's delivery instructions,
the kernel checks readability.
When `qmail-local` runs a
program, the kernel assigns
the same uid to that program.
No extra code required!

Example where q
did badly: excep

djbdns has thous
of conditional br
About half are si
checking for tem

Same for qmail.
Easy to get wron
"if `ipme_init()`
`qmail-remote` w
(fixed in qmail 0.

Easily fixed by be

well:
e.

ago:
am
.

ner uid!

orward:
ad it.
ts!)
s.
e file)

.

Kernel already tracks users.
Kernel already checks readability.
Why not reuse this code?

How qmail delivers to a user:
Start `qmail-local`
under the right uid.

When `qmail-local` reads
the user's delivery instructions,
the kernel checks readability.
When `qmail-local` runs a
program, the kernel assigns
the same uid to that program.
No extra code required!

Example where qmail and
did badly: exception handl

djbdns has thousands
of conditional branches.
About half are simply
checking for temporary err

Same for qmail.
Easy to get wrong: e.g.,
"if `ipme_init()` returned
`qmail-remote` would cont
(fixed in qmail 0.92).

Easily fixed by better langu

Kernel already tracks users.
Kernel already checks readability.
Why not reuse this code?

How qmail delivers to a user:
Start `qmail-local`
under the right uid.

When `qmail-local` reads
the user's delivery instructions,
the kernel checks readability.
When `qmail-local` runs a
program, the kernel assigns
the same uid to that program.
No extra code required!

Example where qmail and djbdns
did badly: exception handling.

djbdns has thousands
of conditional branches.
About half are simply
checking for temporary errors.

Same for qmail.
Easy to get wrong: e.g.,
"if `ipme_init()` returned `-1`,
`qmail-remote` would continue"
(fixed in qmail 0.92).

Easily fixed by better language.

already tracks users.
already checks readability.
ot reuse this code?

mail delivers to a user:
`qmail-local`
the right uid.

`qmail-local` reads
er's delivery instructions,
rnel checks readability.
`qmail-local` runs a
m, the kernel assigns
me uid to that program.
ra code required!

Example where qmail and djbdns did badly: exception handling.

djbdns has thousands
of conditional branches.
About half are simply
checking for temporary errors.

Same for qmail.
Easy to get wrong: e.g.,
"if `ipme_init()` returned `-1`,
`qmail-remote` would continue"
(fixed in qmail 0.92).

Easily fixed by better language.

More s
exampl
identify
reusing
reusing

"Okay,
build a
and wr
But in
we'll st
includi

acks users.
necks readability.
his code?

ers to a user:
`cal`
uid.

`cal` reads
y instructions,
s readability.
`cal` runs a
nel assigns
that program.
quired!

Example where qmail and djbdns
did badly: exception handling.

djbdns has thousands
of conditional branches.
About half are simply
checking for temporary errors.

Same for qmail.
Easy to get wrong: e.g.,
"if `ipme_init()` returned `-1`,
`qmail-remote` would continue"
(fixed in qmail 0.92).

Easily fixed by better language.

More small-code
examples in my c
identifying comm
reusing network t
reusing the filesy

"Okay, we can
build a system w
and write code w
But in a large sy
we'll still have bu
including security

s.
ability.

er:

ons,
ty.
a
s
am.

Example where qmail and djbdns
did badly: exception handling.

djbdns has thousands
of conditional branches.
About half are simply
checking for temporary errors.

Same for qmail.
Easy to get wrong: e.g.,
"if `ipme_init()` returned `-1`,
`qmail-remote` would continue"
(fixed in qmail 0.92).

Easily fixed by better language.

More small-code meta-engi
examples in my qmail pape
identifying common functio
reusing network tools;
reusing the filesystem.

"Okay, we can
build a system with less co
and write code with fewer
But in a large system
we'll still have bugs,
including security holes!"

Example where qmail and djbdns
did badly: exception handling.

djbdns has thousands
of conditional branches.
About half are simply
checking for temporary errors.

Same for qmail.
Easy to get wrong: e.g.,
"if `ipme_init()` returned `-1`,
`qmail-remote` would continue"
(fixed in qmail 0.92).

Easily fixed by better language.

More small-code meta-engineering
examples in my qmail paper:
identifying common functions;
reusing network tools;
reusing the filesystem.

"Okay, we can
build a system with less code,
and write code with fewer bugs.
But in a large system
we'll still have bugs,
including security holes!"

le where qmail and djbdns
dly: exception handling.

has thousands
ditional branches.
half are simply
ng for temporary errors.

for qmail.
o get wrong: e.g.,
`ne_init()` returned `-1`,
`-remote` would continue"
n qmail 0.92).

fixed by better language.

More small-code meta-engineering examples in my qmail paper:
identifying common functions;
reusing network tools;
reusing the filesystem.

"Okay, we can
build a system with less code,
and write code with fewer bugs.
But in a large system
we'll still have bugs,
including security holes!"

<u>Elimina</u>

Can ar
to plac
into *un*

Definit
no mat
no mat
no mat
it cann
user's s

Measur
and me
that re
Note:

qmail and djbdns
tion handling.

sands
anches.
mply
porary errors.

g: e.g.,
returned −1,
would continue"
92).

etter language.

More small-code meta-engineering
examples in my qmail paper:
identifying common functions;
reusing network tools;
reusing the filesystem.

"Okay, we can
build a system with less code,
and write code with fewer bugs.
But in a large system
we'll still have bugs,
including security holes!"

Eliminating trust

Can architect co
to place most of
into *untrusted* p

Definition of "un
no matter what
no matter how b
no matter how m
it cannot violate
user's security re

Measure *trusted*
and meta-engine
that reduce this
Note: progress is

djbdns
ing.

ors.

-1,
tinue"

uage.

More small-code meta-engineering
examples in my qmail paper:
identifying common functions;
reusing network tools;
reusing the filesystem.

"Okay, we can
build a system with less code,
and write code with fewer bugs.
But in a large system
we'll still have bugs,
including security holes!"

Eliminating trusted code

Can architect computer sys
to place most of the code
into *untrusted* prisons.

Definition of "untrusted":
no matter what the code d
no matter how badly it beh
no matter how many bugs
it cannot violate the
user's security requirement

Measure *trusted* code volu
and meta-engineer process
that reduce this volume.
Note: progress is *quantifie*

More small-code meta-engineering
examples in my qmail paper:
identifying common functions;
reusing network tools;
reusing the filesystem.

"Okay, we can
build a system with less code,
and write code with fewer bugs.
But in a large system
we'll still have bugs,
including security holes!"

Eliminating trusted code

Can architect computer systems
to place most of the code
into *untrusted* prisons.

Definition of "untrusted":
no matter what the code does,
no matter how badly it behaves,
no matter how many bugs it has,
it cannot violate the
user's security requirements.

Measure *trusted* code volume,
and meta-engineer processes
that reduce this volume.
Note: progress is *quantified*.

mall-code meta-engineering
es in my qmail paper:
ying common functions;
network tools;
the filesystem.

we can
system with less code,
ite code with fewer bugs.
a large system
till have bugs,
ng security holes!"

Eliminating trusted code

Can architect computer systems
to place most of the code
into *untrusted* prisons.

Definition of "untrusted":
no matter what the code does,
no matter how badly it behaves,
no matter how many bugs it has,
it cannot violate the
user's security requirements.

Measure *trusted* code volume,
and meta-engineer processes
that reduce this volume.
Note: progress is *quantified*.

Warnin
rarely e

Every s
no mat
says it'
This is

qmail a
here: a
I spent
"minim
This di
elimina

meta-engineering
qmail paper:
non functions;
tools;
stem.

ith less code,
with fewer bugs.
stem
ugs,
y holes!"

Eliminating trusted code

Can architect computer systems
to place most of the code
into *untrusted* prisons.

Definition of "untrusted":
no matter what the code does,
no matter how badly it behaves,
no matter how many bugs it has,
it cannot violate the
user's security requirements.

Measure *trusted* code volume,
and meta-engineer processes
that reduce this volume.
Note: progress is *quantified*.

Warning: "Minim
rarely eliminates

Every security m
no matter how p
says it's "minimi
This is not a use

qmail and djbdns
here: almost all
I spent considera
"minimizing privi
This distracted m
eliminating truste

ineering
er:
ons;

de,
bugs.

Eliminating trusted code

Can architect computer systems
to place most of the code
into *untrusted* prisons.

Definition of "untrusted":
no matter what the code does,
no matter how badly it behaves,
no matter how many bugs it has,
it cannot violate the
user's security requirements.

Measure *trusted* code volume,
and meta-engineer processes
that reduce this volume.
Note: progress is *quantified*.

Warning: "Minimizing priv
rarely eliminates trusted co

Every security mechanism,
no matter how pointless,
says it's "minimizing privile
This is not a useful concep

qmail and djbdns did very
here: almost all code is tru
I spent considerable effort
"minimizing privilege"; stu
This distracted me from
eliminating trusted code.

Eliminating trusted code

Can architect computer systems
to place most of the code
into *untrusted* prisons.

Definition of "untrusted":
no matter what the code does,
no matter how badly it behaves,
no matter how many bugs it has,
it cannot violate the
user's security requirements.

Measure *trusted* code volume,
and meta-engineer processes
that reduce this volume.
Note: progress is *quantified*.

Warning: "Minimizing privilege"
rarely eliminates trusted code.

Every security mechanism,
no matter how pointless,
says it's "minimizing privilege."
This is not a useful concept.

qmail and djbdns did very badly
here: almost all code is trusted.
I spent considerable effort
"minimizing privilege"; stupid!
This distracted me from
eliminating trusted code.

ating trusted code

chitect computer systems
e most of the code
ntrusted prisons.

ion of "untrusted":
tter what the code does,
tter how badly it behaves,
tter how many bugs it has,
ot violate the
security requirements.

re *trusted* code volume,
eta-engineer processes
duce this volume.
progress is *quantified*.

Warning: "Minimizing privilege"
rarely eliminates trusted code.

Every security mechanism,
no matter how pointless,
says it's "minimizing privilege."
This is not a useful concept.

qmail and djbdns did very badly
here: almost all code is trusted.
I spent considerable effort
"minimizing privilege"; stupid!
This distracted me from
eliminating trusted code.

What a
security

My fun
The sy
of sour

When t
for data
it does
data fr
to influ

Examp
accoun
I am n
web pa

ed code

mputer systems
the code
risons.

trusted":
the code does,
adly it behaves,
nany bugs it has,
the
quirements.

code volume,
er processes
volume.
s *quantified*.

Warning: "Minimizing privilege"
rarely eliminates trusted code.

Every security mechanism,
no matter how pointless,
says it's "minimizing privilege."
This is not a useful concept.

qmail and djbdns did very badly
here: almost all code is trusted.
I spent considerable effort
"minimizing privilege"; stupid!
This distracted me from
eliminating trusted code.

What *are* the use
security requirem

My fundamental
The system keep
of sources of dat

When the system
for data from sou
it does not allow
data from source
to influence the

Example: When
account statemen
I am not seeing
web pages or em

stems

loes,
naves,
it has,

s.

me,
es

d.

Warning: "Minimizing privilege"
rarely eliminates trusted code.

Every security mechanism,
no matter how pointless,
says it's "minimizing privilege."
This is not a useful concept.

qmail and djbdns did very badly
here: almost all code is trusted.
I spent considerable effort
"minimizing privilege"; stupid!
This distracted me from
eliminating trusted code.

What *are* the user's
security requirements?

My fundamental requireme
The system keeps track
of sources of data.

When the system is asked
for data from source X,
it does not allow
data from source Y
to influence the result.

Example: When I view an
account statement from m
I am not seeing data from
web pages or email messag

Warning: "Minimizing privilege"
rarely eliminates trusted code.

Every security mechanism,
no matter how pointless,
says it's "minimizing privilege."
This is not a useful concept.

qmail and djbdns did very badly
here: almost all code is trusted.
I spent considerable effort
"minimizing privilege"; stupid!
This distracted me from
eliminating trusted code.

What *are* the user's
security requirements?

My fundamental requirement:
The system keeps track
of sources of data.

When the system is asked
for data from source X,
it does not allow
data from source Y
to influence the result.

Example: When I view an
account statement from my bank,
I am not seeing data from other
web pages or email messages.

ng: "Minimizing privilege"
eliminates trusted code.

security mechanism,
tter how pointless,
s "minimizing privilege."
not a useful concept.

and djbdns did very badly
almost all code is trusted.
considerable effort
nizing privilege"; stupid!
stracted me from
ting trusted code.

What *are* the user's
security requirements?

My fundamental requirement:
The system keeps track
of sources of data.

When the system is asked
for data from source X,
it does not allow
data from source Y
to influence the result.

Example: When I view an
account statement from my bank,
I am not seeing data from other
web pages or email messages.

There i
central
of sour
Can an
to elim

Classic
used ve
to trac
VAX V
had <

Minor
suppor
instead

mizing privilege"
trusted code.

echanism,
ointless,
zing privilege."
ful concept.

s did very badly
code is trusted.
ble effort
ilege"; stupid!
ne from
ed code.

What *are* the user's
security requirements?

My fundamental requirement:
The system keeps track
of sources of data.

When the system is asked
for data from source X,
it does not allow
data from source Y
to influence the result.

Example: When I view an
account statement from my bank,
I am not seeing data from other
web pages or email messages.

There is no obsta
centralization an
of source-trackin
Can and should b
to eliminate all b

Classic military T
used very few lin
to track (e.g.) To
VAX VMM Secu
had $<$ 50000 line

Minor programm
support arbitrary
instead of Top S

ilege"
ode.

ege."
t.
badly
usted.

pid!

What *are* the user's
security requirements?

My fundamental requirement:
The system keeps track
of sources of data.

When the system is asked
for data from source X,
it does not allow
data from source Y
to influence the result.

Example: When I view an
account statement from my bank,
I am not seeing data from other
web pages or email messages.

There is no obstacle to
centralization and minimiza
of source-tracking code.
Can and should be small e
to eliminate all bugs.

Classic military TCBs
used very few lines of code
to track (e.g.) Top Secret
VAX VMM Security Kerne
had $<$ 50000 lines of code.

Minor programming proble
support arbitrary source lab
instead of Top Secret etc.

What *are* the user's
security requirements?

My fundamental requirement:
The system keeps track
of sources of data.

When the system is asked
for data from source X,
it does not allow
data from source Y
to influence the result.

Example: When I view an
account statement from my bank,
I am not seeing data from other
web pages or email messages.

There is no obstacle to
centralization and minimization
of source-tracking code.
Can and should be small enough
to eliminate all bugs.

Classic military TCBs
used very few lines of code
to track (e.g.) Top Secret data.
VAX VMM Security Kernel
had $<$ 50000 lines of code.

Minor programming problem to
support arbitrary source labels
instead of Top Secret etc.

are the user's
y requirements?

ndamental requirement:
stem keeps track
ces of data.

the system is asked
a from source X,
not allow
om source Y
uence the result.

le: When I view an
t statement from my bank,
ot seeing data from other
ges or email messages.

There is no obstacle to
centralization and minimization
of source-tracking code.
Can and should be small enough
to eliminate all bugs.

Classic military TCBs
used very few lines of code
to track (e.g.) Top Secret data.
VAX VMM Security Kernel
had $<$ 50000 lines of code.

Minor programming problem to
support arbitrary source labels
instead of Top Secret etc.

"Doesn
already

If I log
the ker
to my
to othe
to files

But if
to anot
through
the ker
that I'm

er's
nents?

requirement:
s track
a.

n is asked
urce X,
e Y
result.

l view an
nt from my bank,
data from other
ail messages.

There is no obstacle to
centralization and minimization
of source-tracking code.
Can and should be small enough
to eliminate all bugs.

Classic military TCBs
used very few lines of code
to track (e.g.) Top Secret data.
VAX VMM Security Kernel
had $<$ 50000 lines of code.

Minor programming problem to
support arbitrary source labels
instead of Top Secret etc.

"Doesn't the UN
already track sou

If I log into a sys
the kernel copies
to my login proc
to other processe
to files I create, e

But if I transfer
to another user's
through the netw
the kernel doesn'
that I'm the sour

ent:

y bank,
other
ges.

There is no obstacle to
centralization and minimization
of source-tracking code.
Can and should be small enough
to eliminate all bugs.

Classic military TCBs
used very few lines of code
to track (e.g.) Top Secret data.
VAX VMM Security Kernel
had $<$ 50000 lines of code.

Minor programming problem to
support arbitrary source labels
instead of Top Secret etc.

"Doesn't the UNIX/Linux
already track sources?"

If I log into a system,
the kernel copies my uid
to my login process,
to other processes I start,
to files I create, etc.

But if I transfer data
to another user's processes
through the network or a f
the kernel doesn't rememb
that I'm the source.

There is no obstacle to
centralization and minimization
of source-tracking code.
Can and should be small enough
to eliminate all bugs.

Classic military TCBs
used very few lines of code
to track (e.g.) Top Secret data.
VAX VMM Security Kernel
had $<$ 50000 lines of code.

Minor programming problem to
support arbitrary source labels
instead of Top Secret etc.

"Doesn't the UNIX/Linux kernel
already track sources?"

If I log into a system,
the kernel copies my uid
to my login process,
to other processes I start,
to files I create, etc.

But if I transfer data
to another user's processes—
through the network or a file—
the kernel doesn't remember
that I'm the source.

is no obstacle to
ization and minimization
ce-tracking code.
d should be small enough
inate all bugs.

military TCBs
ery few lines of code
k (e.g.) Top Secret data.
MM Security Kernel
50000 lines of code.

programming problem to
t arbitrary source labels
of Top Secret etc.

"Doesn't the UNIX/Linux kernel
already track sources?"

If I log into a system,
the kernel copies my uid
to my login process,
to other processes I start,
to files I create, etc.

But if I transfer data
to another user's processes—
through the network or a file—
the kernel doesn't remember
that I'm the source.

Source
is impl
writing
PHP s

All of t
All oth
is also
nonexis

Your la
of lines
thousa
A scre
can vio

acle to
d minimization
g code.
be small enough
bugs.

TCBs
es of code
op Secret data.
rity Kernel
es of code.

ing problem to
source labels
ecret etc.

"Doesn't the UNIX/Linux kernel
already track sources?"

If I log into a system,
the kernel copies my uid
to my login process,
to other processes I start,
to files I create, etc.

But if I transfer data
to another user's processes—
through the network or a file—
the kernel doesn't remember
that I'm the source.

Source tracking t
is implemented b
writing web brow
PHP scripts, etc.

All of this code i
All other code in
is also trusted, th
nonexistent inter

Your laptop has
of lines of trusted
thousands of nov
A screwup anywh
can violate secur

ation

nough

e

data.

l

m to

bels

"Doesn't the UNIX/Linux kernel
already track sources?"

If I log into a system,
the kernel copies my uid
to my login process,
to other processes I start,
to files I create, etc.

But if I transfer data
to another user's processes—
through the network or a file—
the kernel doesn't remember
that I'm the source.

Source tracking today
is implemented by program
writing web browsers, mail
PHP scripts, etc.

All of this code is trusted.
All other code in these pro
is also trusted, thanks to
nonexistent internal partiti

Your laptop has tens of mi
of lines of trusted code wri
thousands of novice progra
A screwup anywhere in tha
can violate security require

"Doesn't the UNIX/Linux kernel
already track sources?"

If I log into a system,
the kernel copies my uid
to my login process,
to other processes I start,
to files I create, etc.

But if I transfer data
to another user's processes—
through the network or a file—
the kernel doesn't remember
that I'm the source.

Source tracking today
is implemented by programmers
writing web browsers, mail clients,
PHP scripts, etc.

All of this code is trusted.
All other code in these programs
is also trusted, thanks to
nonexistent internal partitioning.

Your laptop has tens of millions
of lines of trusted code written by
thousands of novice programmers.
A screwup anywhere in that code
can violate security requirements.

n't the UNIX/Linux kernel
y track sources?"

into a system,
rnel copies my uid
login process,
er processes I start,
I create, etc.

l transfer data
ther user's processes—
h the network or a file—
rnel doesn't remember
m the source.

Source tracking today
is implemented by programmers
writing web browsers, mail clients,
PHP scripts, etc.

All of this code is trusted.
All other code in these programs
is also trusted, thanks to
nonexistent internal partitioning.

Your laptop has tens of millions
of lines of trusted code written by
thousands of novice programmers.
A screwup anywhere in that code
can violate security requirements.

"Teach
how to

No, no
If every
is writi
then th
far too

We nee
with fa
Enforce
so that
don't h

IX/Linux kernel
rces?"

stem,
my uid
ess,
es I start,
etc.

data
processes—
ork or a file—
t remember
rce.

Source tracking today
is implemented by programmers
writing web browsers, mail clients,
PHP scripts, etc.

All of this code is trusted.
All other code in these programs
is also trusted, thanks to
nonexistent internal partitioning.

Your laptop has tens of millions
of lines of trusted code written by
thousands of novice programmers.
A screwup anywhere in that code
can violate security requirements.

"Teach every pro
how to write secu

No, no, no!
If every programm
is writing trusted
then the system
far too much tru

We need new sys
with far less trus
Enforce security
so that typical pr
don't have to wo

kernel

s—
ile—
er

Source tracking today
is implemented by programmers
writing web browsers, mail clients,
PHP scripts, etc.

All of this code is trusted.
All other code in these programs
is also trusted, thanks to
nonexistent internal partitioning.

Your laptop has tens of millions
of lines of trusted code written by
thousands of novice programmers.
A screwup anywhere in that code
can violate security requirements.

"Teach every programmer
how to write secure code."

No, no, no!
If every programmer
is writing trusted code
then the system has
far too much trusted code.

We need new systems
with far less trusted code.
Enforce security in TCB
so that typical programmer
don't have to worry about

Source tracking today
is implemented by programmers
writing web browsers, mail clients,
PHP scripts, etc.

All of this code is trusted.
All other code in these programs
is also trusted, thanks to
nonexistent internal partitioning.

Your laptop has tens of millions
of lines of trusted code written by
thousands of novice programmers.
A screwup anywhere in that code
can violate security requirements.

"Teach every programmer
how to write secure code."

No, no, no!
If every programmer
is writing trusted code
then the system has
far too much trusted code.

We need new systems
with far less trusted code.
Enforce security in TCB
so that typical programmers
don't have to worry about it.

tracking today
emented by programmers
web browsers, mail clients,
cripts, etc.

this code is trusted.
er code in these programs
trusted, thanks to
stent internal partitioning.

aptop has tens of millions
s of trusted code written by
nds of novice programmers.
wup anywhere in that code
olate security requirements.

"Teach every programmer
how to write secure code."

No, no, no!
If every programmer
is writing trusted code
then the system has
far too much trusted code.

We need new systems
with far less trusted code.
Enforce security in TCB
so that typical programmers
don't have to worry about it.

Imagin

Alice's
TCB a
process

Process
TCB a
file as
(and re
labelle

Joe's p
and an
Process
/Joe/A

today
by programmers
wsers, mail clients,
r

s trusted.
these programs
hanks to
nal partitioning.

tens of millions
d code written by
vice programmers.
here in that code
ity requirements.

"Teach every programmer
how to write secure code."

No, no, no!
If every programmer
is writing trusted code
then the system has
far too much trusted code.

We need new systems
with far less trusted code.
Enforce security in TCB
so that typical programmers
don't have to worry about it.

Imagine a TCB t

Alice's process re
TCB automatica
process as /Alice

Process creates a
TCB automatica
file as /Alice/Bo
(and refuses to t
labelled only /Al

Joe's process rea
and another file f
Process then has
/Joe/Alice/Bob;

nmers
clients,

grams

oning.

llions
tten by
mmers.
at code
ments.

"Teach every programmer
how to write secure code."

No, no, no!
If every programmer
is writing trusted code
then the system has
far too much trusted code.

We need new systems
with far less trusted code.
Enforce security in TCB
so that typical programmers
don't have to worry about it.

Imagine a TCB tracking so

Alice's process reads Bob's
TCB automatically labels
process as /Alice/Bob.

Process creates a file.
TCB automatically labels
file as /Alice/Bob
(and refuses to touch a file
labelled only /Alice).

Joe's process reads the file
and another file from Char
Process then has two label
/Joe/Alice/Bob; /Joe/Cha

"Teach every programmer
how to write secure code."

No, no, no!
If every programmer
is writing trusted code
then the system has
far too much trusted code.

We need new systems
with far less trusted code.
Enforce security in TCB
so that typical programmers
don't have to worry about it.

Imagine a TCB tracking sources.

Alice's process reads Bob's file.
TCB automatically labels
process as /Alice/Bob.

Process creates a file.
TCB automatically labels
file as /Alice/Bob
(and refuses to touch a file
labelled only /Alice).

Joe's process reads the file
and another file from Charlie.
Process then has two labels:
/Joe/Alice/Bob; /Joe/Charlie.

n every programmer
write secure code."

, no!
y programmer
ng trusted code
he system has
much trusted code.

ed new systems
r less trusted code.
e security in TCB
t typical programmers
have to worry about it.

Imagine a TCB tracking sources.

Alice's process reads Bob's file.
TCB automatically labels
process as /Alice/Bob.

Process creates a file.
TCB automatically labels
file as /Alice/Bob
(and refuses to touch a file
labelled only /Alice).

Joe's process reads the file
and another file from Charlie.
Process then has two labels:
/Joe/Alice/Bob; /Joe/Charlie.

A web-
that re
and fro
will hav
TCB w
to writ

Solutio
in the
one pro

Solutio
separat
inside t

ogrammer
ure code."

mer
code
has
sted code.

stems
ted code.
in TCB
rogrammers
rry about it.

Imagine a TCB tracking sources.

Alice's process reads Bob's file.
TCB automatically labels
process as /Alice/Bob.

Process creates a file.
TCB automatically labels
file as /Alice/Bob
(and refuses to touch a file
labelled only /Alice).

Joe's process reads the file
and another file from Charlie.
Process then has two labels:
/Joe/Alice/Bob; /Joe/Charlie.

A web-browsing
that reads from
and from nytime
will have both la
TCB won't allow
to write under ju

Solution 1: Rewr
in the classic UN
one process for e

Solution 2: Trac
separately for eac
inside the web-br

Imagine a TCB tracking sources.

Alice's process reads Bob's file.
TCB automatically labels
process as /Alice/Bob.

Process creates a file.
TCB automatically labels
file as /Alice/Bob
(and refuses to touch a file
labelled only /Alice).

Joe's process reads the file
and another file from Charlie.
Process then has two labels:
/Joe/Alice/Bob; /Joe/Charlie.

A web-browsing process
that reads from mbna.com
and from nytimes.com
will have both labels.
TCB won't allow process
to write under just one lab

Solution 1: Rewrite browse
in the classic UNIX style,
one process for each page.

Solution 2: Track sources
separately for each variable
inside the web-browsing pr

Imagine a TCB tracking sources.

Alice's process reads Bob's file.
TCB automatically labels
process as /Alice/Bob.

Process creates a file.
TCB automatically labels
file as /Alice/Bob
(and refuses to touch a file
labelled only /Alice).

Joe's process reads the file
and another file from Charlie.
Process then has two labels:
/Joe/Alice/Bob; /Joe/Charlie.

A web-browsing process
that reads from mbna.com
and from nytimes.com
will have both labels.
TCB won't allow process
to write under just one label.

Solution 1: Rewrite browser
in the classic UNIX style,
one process for each page.

Solution 2: Track sources
separately for each variable
inside the web-browsing process.

e a TCB tracking sources.

process reads Bob's file.
utomatically labels
s as /Alice/Bob.

s creates a file.
utomatically labels
/Alice/Bob
efuses to touch a file
d only /Alice).

process reads the file
other file from Charlie.
s then has two labels:
Alice/Bob; /Joe/Charlie.

A web-browsing process
that reads from mbna.com
and from nytimes.com
will have both labels.
TCB won't allow process
to write under just one label.

Solution 1: Rewrite browser
in the classic UNIX style,
one process for each page.

Solution 2: Track sources
separately for each variable
inside the web-browsing process.

Suppos
receive
the `mi`
with `ww`

Packet
TCB a
a `micr`

Packet-
extract
informa

TCB a
the `mi`
to deri
the stri

tracking sources.

eads Bob's file.
lly labels
e/Bob.

a file.
lly labels
b
ouch a file
ice).

ds the file
from Charlie.
two labels:
/Joe/Charlie.

A web-browsing process
that reads from mbna.com
and from nytimes.com
will have both labels.
TCB won't allow process
to write under just one label.

Solution 1: Rewrite browser
in the classic UNIX style,
one process for each page.

Solution 2: Track sources
separately for each variable
inside the web-browsing process.

Suppose a DNS
receives a packet
the `microsoft.`
with `www.google`

Packet is a string
TCB attaches to
a `microsoft.co`

Packet-parsing co
extracts `www.goo`
information from

TCB automatica
the `microsoft.`
to derived variab
the string `www.g`

ources.

file.

lie.

s:

arlie.

A web-browsing process
that reads from mbna.com
and from nytimes.com
will have both labels.
TCB won't allow process
to write under just one label.

Solution 1: Rewrite browser
in the classic UNIX style,
one process for each page.

Solution 2: Track sources
separately for each variable
inside the web-browsing process.

Suppose a DNS cache
receives a packet from
the `microsoft.com` server
with `www.google.com` dat

Packet is a string.
TCB attaches to the string
a `microsoft.com` source l

Packet-parsing code
extracts `www.google.com`
information from the packe

TCB automatically copies
the `microsoft.com` source
to derived variables such as
the string `www.google.co`

A web-browsing process
that reads from mbna.com
and from nytimes.com
will have both labels.
TCB won't allow process
to write under just one label.

Solution 1: Rewrite browser
in the classic UNIX style,
one process for each page.

Solution 2: Track sources
separately for each variable
inside the web-browsing process.

Suppose a DNS cache
receives a packet from
the `microsoft.com` servers
with `www.google.com` data.

Packet is a string.
TCB attaches to the string
a `microsoft.com` source label.

Packet-parsing code
extracts `www.google.com`
information from the packet.

TCB automatically copies
the `microsoft.com` source label
to derived variables such as
the string `www.google.com`.

-browsing process
ads from mbna.com
om nytimes.com
ve both labels.
won't allow process
e under just one label.

n 1: Rewrite browser
classic UNIX style,
ocess for each page.

n 2: Track sources
tely for each variable
the web-browsing process.

Suppose a DNS cache
receives a packet from
the `microsoft.com` servers
with `www.google.com` data.

Packet is a string.
TCB attaches to the string
a `microsoft.com` source label.

Packet-parsing code
extracts `www.google.com`
information from the packet.

TCB automatically copies
the `microsoft.com` source label
to derived variables such as
the string `www.google.com`.

The ca
in a big

Post-pa
tries to
in the a

Cache
`.googl`
are allo
`www.go`

TCB e
sees `mi`
refuses

process
mbna.com
s.com
bels.
process
st one label.

rite browser
IX style,
ach page.

k sources
ch variable
rowsing process.

Suppose a DNS cache
receives a packet from
the `microsoft.com` servers
with `www.google.com` data.

Packet is a string.
TCB attaches to the string
a `microsoft.com` source label.

Packet-parsing code
extracts `www.google.com`
information from the packet.

TCB automatically copies
the `microsoft.com` source label
to derived variables such as
the string `www.google.com`.

The cache remem
in a big associati

Post-packet-parsi
tries to store `www`
in the associative

Cache policy: on
`.google.com`, `.w`
are allowed to st
`www.google.com`

TCB enforces thi
sees `microsoft.`
refuses `www.goog`

el.

er

e

ocess.

Suppose a DNS cache
receives a packet from
the `microsoft.com` servers
with `www.google.com` data.

Packet is a string.
TCB attaches to the string
a `microsoft.com` source label.

Packet-parsing code
extracts `www.google.com`
information from the packet.

TCB automatically copies
the `microsoft.com` source label
to derived variables such as
the string `www.google.com`.

The cache remembers infor
in a big associative array.

Post-packet-parsing code
tries to store `www.google.`
in the associative array.

Cache policy: only root, `.c`
`.google.com`, `.www.goog`
are allowed to store
`www.google.com` informat

TCB enforces this policy:
sees `microsoft.com` label
refuses `www.google.com` d

Suppose a DNS cache
receives a packet from
the `microsoft.com` servers
with `www.google.com` data.

Packet is a string.
TCB attaches to the string
a `microsoft.com` source label.

Packet-parsing code
extracts `www.google.com`
information from the packet.

TCB automatically copies
the `microsoft.com` source label
to derived variables such as
the string `www.google.com`.

The cache remembers information
in a big associative array.

Post-packet-parsing code
tries to store `www.google.com`
in the associative array.

Cache policy: only root, `.com`,
`.google.com`, `.www.google.com`
are allowed to store
`www.google.com` information.

TCB enforces this policy:
sees `microsoft.com` label,
refuses `www.google.com` data.

se a DNS cache
s a packet from
crosoft.com servers
ww.google.com data.

is a string.
ttaches to the string
rosoft.com source label.

-parsing code
s www.google.com
ation from the packet.

utomatically copies
crosoft.com source label
ved variables such as
ing www.google.com.

The cache remembers information
in a big associative array.

Post-packet-parsing code
tries to store www.google.com
in the associative array.

Cache policy: only root, .com,
.google.com, .www.google.com
are allowed to store
www.google.com information.

TCB enforces this policy:
sees microsoft.com label,
refuses www.google.com data.

How m
for a T
source-
against

How m
in a TC
Note:
technic

If code
and bu
then w
that so

cache
from
com servers
e.com data.

g.
the string
om source label.

ode
ogle.com
the packet.

lly copies
com source label
les such as
oogle.com.

The cache remembers information
in a big associative array.

Post-packet-parsing code
tries to store www.google.com
in the associative array.

Cache policy: only root, .com,
.google.com, .www.google.com
are allowed to store
www.google.com information.

TCB enforces this policy:
sees microsoft.com label,
refuses www.google.com data.

How much code
for a TCB that e
source-tracking p
against all other

How many bugs
in a TCB of this
Note: Can afford
techniques to red

If code volume is
and bug rate is s
then we will be c
that sources are

rs
a.

g
abel.

et.

e label
s
m.

The cache remembers information
in a big associative array.

Post-packet-parsing code
tries to store `www.google.com`
in the associative array.

Cache policy: only root, `.com`,
`.google.com`, `.www.google.com`
are allowed to store
`www.google.com` information.

TCB enforces this policy:
sees `microsoft.com` label,
refuses `www.google.com` data.

How much code is required
for a TCB that enforces
source-tracking policy
against all other code?

How many bugs do we exp
in a TCB of this size?
Note: Can afford expensive
techniques to reduce bug r

If code volume is small enc
and bug rate is small enou
then we will be confident
that sources are tracked.

The cache remembers information
in a big associative array.

Post-packet-parsing code
tries to store `www.google.com`
in the associative array.

Cache policy: only root, `.com`,
`.google.com`, `.www.google.com`
are allowed to store
`www.google.com` information.

TCB enforces this policy:
sees `microsoft.com` label,
refuses `www.google.com` data.

How much code is required
for a TCB that enforces
source-tracking policy
against all other code?

How many bugs do we expect
in a TCB of this size?
Note: Can afford expensive
techniques to reduce bug rate.

If code volume is small enough,
and bug rate is small enough,
then we will be confident
that sources are tracked.