Setiri:
Advances in Trojan Technology

Roelof Temmingh & Haroon Meer
BlackHat USA
Las Vegas
2002
Schedule

Introduction
Why Trojans?
Brief History of Trojans & Covert Channels

The Hybrid model
Setiri: Advances in Trojan Technology
Demonstration

Taking it further
Possible fixes
Introduction

SensePost

The speakers

Objective of presentation
Why Trojans?

Profile of Trojan users

Real criminals…
… don’t write buffer overflows

The weirdness of the industry

Examples
Brief History of Trojans & Covert Tunnels

Trojans
From Quick Thinking Greeks …
to Quick Thinking Geeks

Tunnels
Covert Channels
Trojans..

Valid IP – No Filters

Valid IP – Stateless Filters

Private Addresses – Stateful Filters

Private
+ Stateful
+ IDS + Personal Firewalls
+ Content Checking
+ ...
Trojans (Valid IP – No Filters)

“get real..”
Trojans (Valid IP – Stateless Filter)

Dial Home Trojans

Random Ports / Open Ports / High Ports
[cDc]

ACK Tunneling
[Arne Vidstrom]
Trojans (Stateful Filters)

Gbot
Rattler
Brief History of Trojans & Covert Tunnels

Trojans
From Quick Thinking Greeks …
to Quick Thinking Geeks

Tunnels
Covert Channels
Tunnels & Covert Channels

1985 – TSC Definition “Covert Channels”

1996 – Phrack Magazine – LOKI

1998 – RWWWShell – THC

1999 - HTTPTUNNEL – GNU

2000 - FireThru - Firethru
Conventional Trojans & how they fail

Stateful firewall & IDS
  Direct model
  Direct model with network tricks
  ICMP tunneling
  ACK tunneling

Properly configured stateful firewall
  IRC agents +

Authentication proxy
  HTTP tunnel ++

Personal firewall & Advanced Proxy
  HTTP tunnel with Authentication +++
Hybrid model: “GatSlag”

Combination between covert Tunnel and Trojan

Defenses mechanisms today:

Packet filters (stateful) / NAT Authentication Proxies
Intrusion detection systems
Personal firewalls
Content/protocol checking
Biometrics/Token Pads/One time passwords
Encryption
A typical network

Controller

Internet

Screening Internet Router

NAT firewall (stateful)

IDS/virus scanner

Authentication Proxy

unrouted net

Content checking Firewall

unrouted net

Victim PC with personal FW
How GatSlag worked

Reverse connection
HTTP covert tunnel

Microsoft Internet Explorer as transport

Controls IE via OLE
Encapsulate in IE, not HTTP
Receive commands in title of web page
Receive encoded data as plain text in body of web page
Send data with POST request
Send alive signals with GET request
Why GatSlag worked

- Integration of client with MS Proxy
- NTLM authentication
- SSL capable
- Registry changes
- Personal firewalls
- Just another browser
- Platform independent
- IE on every desktop

Specify Controller
Via public web page – the MASTERS site
How GatSlag worked II

Creates invisible browser

Find controller at MASTER
Send request to Controller
If no Controller && retry>7, go to MASTER
Receive reply
Parse reply:
  + Upload file()
  + Download file
  + Execute command
Loop
Problems with *Gatslag*

The Controller’s IP can be obtained!
Handling of multiple instances
GUI support
Controller needed to be online
Batch commands
Command history
Multiple controllers
Upload facility not efficient
Platform support
Stability

Session level tunneling
Setiri: Advances in Trojan Technology

Design notes:

Web site contains instructions
CGIs to create new instruction

Controller’s interface:
– EXEC (DOS commands, various)
– TX (File upload via upload CGI)
– RX (File download, UUencode)

Directory structure – each instance
Trojan “surfs” to web site – just a normal user would
Setiri: Advances in Trojan Technology II

Anonymity

Problems with normal proxies
Already using a proxy
Proxy logs

“Cleaners” provide anonymity
“In browser proxy” – Anonymizer
Trojan -> Cleaner: SSL
Cleaner -> Controller: SSL

Challenges:
Browser history
Temporary files
Why defenses fail

Firewalls (stateful/NAT)
Configured to allow user or proxy out

Content level & IDS
Looks like valid HTTP requests & replies
Files downloaded as uuencoded text in web pages
No data or ports to lock on to
SSL provides encryption

Personal firewalls
IE valid application
Configured to allow browsing

Authentication proxies
User surf the web
Taking it further

Session level tunneling
Flow control challenges

How this is different from HTTP tunneling

A browser is not a socket
No select on browser
Train model

The Controller side
Cannot “send”
Buffering of data at Controller

The Trojan side
Multi-part POSTs

Multiple connections (HTTP)
True network level tunneling
Solving the dilemma

Delivery

White listing

User education

AV, personal firewalls

Should you allow everyone to surf the ‘net?'
Conclusion

Awareness

Our motivation