A Stateful Inspection of FireWall-1

Thomas Lopatic, John McDonald
TÜV data protect GmbH
tl@dataprotect.com, jm@dataprotect.com

Dug Song
CITI at the University of Michigan
dugsong@umich.edu
Overview

- Architecture of FireWall-1
- Attacking the firewall’s state I
- FWZ encapsulation
- Attacking the firewall’s state II
- Attacking authentication between firewall modules
- Hardening FireWall-1
- The big picture
Topology

Victim network

Hostile network

Problems in Inspection

• Unreliable / unauthenticated input
• Layering restrictions on inspection
• Layering violations in inspection
• Ambiguous end-to-end semantics
Example: Airport Security

- Unreliable / unauthenticated input
  Examining baggage tags

- Layering restrictions on inspection
  Examining shape, size, weight

- Layering violations in inspection
  Parallelizing bag content inspection

- Ambiguous end-to-end semantics
  Checking for known contraband
Classification of the Attacks

- Unreliable / unauthenticated input
  - TCP fastmode
- Layering restrictions on inspection
  - FWZ VPN encapsulation
- Layering violations in inspection
  - FTP data connection handling
  - unidirectional TCP data flow
  - RSH error connection handling
- Ambiguous end-to-end semantics
  - Parsing of FTP “PORT” commands
FireWall-1 Modules

Port 258/TCP
Management module

Port 256/TCP
Authentication methods
S/Key, FWN1, FWA1

Security policy, status, logs

GUI

Filter module

Filter module

Filter module
Inter-Module Protocol

Management module

Version →

Version ←

Command →

IP addresses →

IP addresses ←

Required authentication

Authentication

Arguments, Result

Filter module
S/Key Authentication

\[ \text{Hash}_n(x) = \text{Hash}(\text{Hash}(\ldots \text{Hash}(x))) = \text{Hash}(\text{Hash}_{n-1}(x)) \]

\( n \) times

Seed \( x \)
(password hash)

<table>
<thead>
<tr>
<th>Index = 99</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash_{99}(x)</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Index = 1</td>
</tr>
<tr>
<td>Hash_1(x)</td>
</tr>
<tr>
<td>Calculate seed ( y ), Hash_{100}(y)</td>
</tr>
<tr>
<td>Hash_{100}(x)</td>
</tr>
</tbody>
</table>

- \( y = \text{MakeSeed}(\text{time}(\text{NULL})) \)
- Attack: brute force

FWN1 Authentication

- Shared key $K$ ("fw putkey")
- Attack: choose $R_2 = R_1$, so that $S_2 = S_1$
FWA1 Authentication

Random number $R_1$

$S_1 = \text{Hash}(R_1 + K)$

Random number $R_2$

$S_2 = \text{Hash}((R_1 \land R_2) + K)$

• Shared key $K$ (“fw putkey”)
• Attack: choose $R_2 = 0$, so that
  • $R_1 \land R_2 = R_1$ and
  • $S_2 = \text{Hash}((R_1 \land R_2) + K) = \text{Hash}(R_1 + K) = S_1$
• To be solved: encryption
Stateful Inspection I

Stateful Inspection II

- UDP “connections”
  - from a client, port C
  - to a server, port S + wildcard port
  - <s-address, s-port, d-address, d-port, protocol>
Stateful Inspection III

"PORT 192.168.0,2,4,36"

FTP server 172.16.0.2

FTP client 192.168.0.2

"PASV"

"227 ... (172,16,0,2,4,36)"

FTP server 172.16.0.2

FTP client 192.168.0.2
Fastmode Services

- non-SYN packets accepted
  - Source port = fastmode service
  - Destination port = fastmode service
- Stealth scanning (FINs, ...)

FTP “PORT” Parsing

“PORT 172,16,0,258,p1,p2”

172.16.0.2

172.16.1.2

data connection

Application: bounce attack

“PORT 172,16,1349632,2,p1,p2”

1349632 = 65536 * (192 - 172) + 256 * (168 - 16)

172.16.0.2

192.168.0.2

FTP “PASV” Handling

- Advertise small Maximal Segment Size
- Server replies split

```
172.16.0.2

"XXXXXXXXXXXXXXXXXX227 (172,16,0,2,128,7)"

500 Invalid command given
en: XXXXXXXXXXXXXXXXXX
227 (172,16,0,2,128,7)

192.168.0.2
```
One-way Connections I

TCP header + payload

DROP

ACCEPT

established one-way connection

One-way Connections II

172.16.0.2

open one-way connection

datagram A

datagram B

open one-way connection

retransmission of B

[...]

192.168.0.2
FWZ Encapsulation I

1. original d-address, original protocol

2. d-address = firewall, protocol = 94

- VPN tunneling protocol
- Decapsulation without decryption or authentication
- Cannot be disabled
FWZ Encapsulation II

Key to spoofing attacks

s-addr = 10.0.0.1
d-addr = 194.221.6.19
d-addr = 131.159.1.1

IP header
encapsulation info
Fake “PORT” Commands

- IP header: s-addr = 172.16.0.2, d-addr = 192.168.0.1
- TCP header + payload: "PORT 172,16,0,2,128,7"
- Encapsulation info: d-addr = 192.168.0.2

FTP client 172.16.0.2 → 192.168.0.1
fake “PORT” packet
192.168.0.1 → 192.168.0.2


22
RSH Error Connections I

- `<172.16.0.2, 1024, 192.168.0.2, 514, 6>` in “connections”
- `<172.16.0.2, 1025, 192.168.0.2, magic, 6>` in “pending”
- Reversed matching

T. Lopatic, J. McDonald, D. Song, ”A Stateful Inspection of FireWall-1”, Black Hat Briefings 2000
RSH Error Connections II

SYN

• s-addr:s-port
• d-addr:magic
• seq + 1

• 172.16.0.2:1024
• 192.168.0.2:magic
• 250001

packet #2
(port info)

• s-addr:error-port
• d-addr:magic
• protocol

• 172.16.0.2:1025
• 192.168.0.2:magic
• 6 (TCP)

seq = 5

• 172.16.0.2:32775
• 192.168.0.2:magic
• 6 = seq + 1 = TCP
Fake UDP Requests

<table>
<thead>
<tr>
<th></th>
<th>s-addr = 172.16.0.2</th>
<th>d-addr = 192.168.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s-port = 161</td>
<td>d-port = 53</td>
</tr>
<tr>
<td></td>
<td>d-addr = 192.168.0.2</td>
<td></td>
</tr>
</tbody>
</table>

IP header

UDP header

encapsulation info

DNS client 172.16.0.2

192.168.0.1

192.168.0.2

fake DNS request
FWZ Encapsulation III

Key to non-routable addresses

s-addr = 131.159.1.1
d-addr = 194.221.6.19
d-addr = 10.0.0.1

IP header
encapsulation info
Anti-Spoofing Protection I

1. 

<table>
<thead>
<tr>
<th></th>
<th>s-addr = 192.168.0.1</th>
<th>d-addr = 192.168.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s-port = 161</td>
<td>d-port = 53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>d-addr = 192.168.0.2</td>
</tr>
</tbody>
</table>

2. 

<table>
<thead>
<tr>
<th></th>
<th>s-addr = 192.168.0.2</th>
<th>d-addr = 192.168.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s-port = any</td>
<td>d-port = 161</td>
</tr>
</tbody>
</table>

1. fake DNS request

2. tunnel to firewall

192.168.0.1

192.168.0.2
Anti-Spoofing Protection II

1. fake DNS request

<table>
<thead>
<tr>
<th>s-addr = 224.0.0.1</th>
<th>d-addr = 192.168.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>s-port = 161</td>
<td>d-port = 53</td>
</tr>
<tr>
<td>d-addr = 192.168.0.2</td>
<td></td>
</tr>
</tbody>
</table>

2. tunnel to firewall

<table>
<thead>
<tr>
<th>s-addr = 192.168.0.2</th>
<th>d-addr = 192.168.0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>s-port = 53</td>
<td>d-port = 161</td>
</tr>
<tr>
<td>d-addr = 224.0.0.1</td>
<td></td>
</tr>
</tbody>
</table>
Hardening I

- Disable implicit rules
  - DNS
  - control connections
  - ICMP

- Restrictive access rules
  - no “any” sources or destinations
  - deny broadcast / multicast addresses
  - “minimal privilege”

- Properly configure anti-spoofing mechanism

- Filter protocol 94 (e.g. IP Filter)
Hardening II

• Different (virtual) IP addresses for public services
• Restrict control connections
  • FWA1 authentication
  • VPN technology
  • **never** use “127.0.0.1: */none”
• More than one line of defense!
Fixes by Check Point

Solutions by Check Point available today at

http://www.checkpoint.com/techsupport
Thanks.

Thomas Lopatic
tl@dataprotect.com

John McDonald
jm@dataprotect.com

Dug Song
dugsong@umich.edu