Improved Port Knocking with Strong Authentication

Rennie deGraaf, John Aycock, and Michael Jacobson, Jr.

Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
Overview

1) Usefulness of port knocking
2) How port knocking works
3) Problems with existing port knocking systems
4) Our improvements on existing systems
5) Areas for further work
Network Access Authentication

- Any service exposed to a public network can be attacked
- Limiting access by address is not adequate
- Limiting access by user requires authentication
Authentication is traditionally left up to the application

But...

- Some applications have no authentication
- Flaws in authentication can allow it to be bypassed
Attacks on Network Authentication

--- kremvax.cccp.su ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000 ms
[root@apollo ~]# nmap -sS -O kremvax.cccp.su

Starting nmap 2.3BETA13 (http://www.insecure.org/nmap/) at 2000-01-21 07:03 PST
Interesting ports on kremvax (209.151.245.6):
(The 1660 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
22/tcp open  ssh
23/tcp open  telnet
80/tcp open  http
Device type: general purpose
Running: Sun Solaris 2.4
Uptime 1865.356 days (since Mon Dec 12 22:30:14 1994)

Nmap finished: 1 IP address (1 host up) scanned in 44.631 seconds
[root@apollo ~]# telnet
telnet> environ define TTYPROMPT foobar
telnet> o kremvax.cccp.su

SunOS 5.4

root
[root@kremvax ~]#
Attacks on Network Authentication

```
Starting nmap V. 2.54BETA25
Insufficient responses for TCP sequencing (3), OS detection may be less
accurate
Interesting ports on 10.2.2.2:
(The 1539 ports scanned but not shown below are in state: closed)
Port  State  Service
22/tcp  open  ssh

No exact OS matches for host

Nmap run completed -- 1 IP address (1 host up) scanned
# sshnuke 10.2.2.2 -rootpw="Z10N0101"
Connecting to 10.2.2.2:ssh ... successful.
Attempting to exploit SSHv1 CRC32 ... successful.
Resetting root password to "Z10N0101".
System open: Access Level <9>
# ssh 10.2.2.2 -1 root
root@10.2.2.2's password:

RE-Control> disable qml nodes 21 - 48
```

Image from The Matrix Reloaded, copyright 2003, Warner Bros.
Attacks on Network Authentication

Images copyright 1999, CNN
IP-level Authentication for Firewalls

- Defense in depth
- Stop-gap security measure for services with known unpatched vulnerabilities
- Wrapper for services without built-in authentication
- Makes service invisible to port scans
IP-level Authentication for Firewalls

```
64 bytes from 209.151.245.6: icmp_seq=1 ttl=61 time=196 ms

--- kremvax.cccp.su ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000 ms
[root@apollo ~]# nmap -sS -o kremvax.cccp.su

Starting nmap 2.3BETA13 ( http://www.insecure.org/nmap/ ) at 2000-01-21 07:03 PST
Interesting ports on kremvax (209.151.245.6):
(The 1662 ports scanned but not shown below are in state: closed)
PORT       STATE     SERVICE
80/tcp      open     http
Device type: general purpose
Running: Sun Solaris 2.4
Uptime 1865.356 days (since Mon Dec 12 22:30:14 1994)

Nmap finished: 1 IP address (1 host up) scanned in 44.631 seconds
[root@apollo ~]# 
```
Requirements for Firewall Authentication

- Strong authentication
- Resistance to traffic interception and modification
- Interoperability with existing systems
- Low resource demands
- Simplicity
- Stealth
Port Knocking

- Information is encoded as a sequence of TCP or UDP port numbers within a range
- Clients send empty packets to these ports
- Server watches for packets sent to these ports, decodes information, and performs some action
Port Knocking

Send a specific sequence of probes...
Port Knocking

Send a specific sequence of probes...

Server

<table>
<thead>
<tr>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>•</td>
</tr>
<tr>
<td>•</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>•</td>
</tr>
<tr>
<td>•</td>
</tr>
</tbody>
</table>

Firewall

Client

port knock client

sshda
Port Knocking

Send a specific sequence of probes...

Port knock client

Rennie deGraaf, University of Calgary

ACSAC 2005
Port Knocking

Server

<table>
<thead>
<tr>
<th>sshd</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Firewall

Client

port knock client

Send a specific sequence of probes...

Send a specific sequence of probes...
Port Knocking

Send a specific sequence of probes...
Port Knocking

Server

Firewall

Client

sshd

ssh

1
2
•
•
•

... and a port is opened in the firewall
Other Work on Port Knocking

PORTKNOCKING - A system for stealthy authentication across closed ...
Port knocking can be used whenever there is a need to transfer information across closed ports. The port knock daemon can be implemented to respond in any ...
www.portknocking.org/ - 19k - 29 Nov 2005 - Cached - Similar pages

Port Knocking | Linux Journal
Third, because the authentication is built into the port knock sequence, ...
I learned a lot about portknocking. That helps me to build a better website ...
www.linuxjournal.com/article/6811 - 71k - 30 Nov 2005 - Cached - Similar pages

netsecurity.about.com/cs/generalsecurity/a/aa03200...
Similar pages

:: knock - Wiki Index ::
knockd - a port-knocking server ... knockd is a port-knock server. It listens to all traffic on an ethernet (or PPP) interface, looking for special "knock" ...
www.zeroflux.org/cgi-bin/cvstrac/knock/wiki - 14k - Cached - Similar pages

NewsForge | A critique of port knoccking
A critique of port knocking – article related to Security and Networking and
Problems with Existing Systems

- Plain-text authentication
- Broken cryptography
- Network Address Translators
- Sensitive to packet delivery order
- No association between authentication and connection
Enhancements to Port Knocking

- Challenge-response authentication that works even if the client is NATed
- Efficient encoding techniques that allow packets to be re-ordered on delivery
NAT-Aware Unilateral Authentication

- Variant on ISO two-pass unilateral authentication
- Uses server as an identity oracle for client
- The same idea also works for mutual authentication
Port Knocking with Strong Authentication

1. Request sequence

Server

Firewall

Client

sshd

1

2

port knock client

Request sequence
Port Knocking with Strong Authentication

Server

Firewall

Client

sshda

1
2
•
•
•

port knock client

Request sequence

1

2

Rennie deGraaf, University of Calgary

ACSAC 2005
Port Knocking with Strong Authentication

Request sequence

Server | Firewall | Client

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>sshd</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

port knock client

Rennie deGraaf, University of Calgary

ACSAC 2005
Port Knocking with Strong Authentication

Request sequence

Server

Firewall

Client

sshd

1

2

•

•

•

22

1

3

2

4

port knock client

Request sequence
Port Knocking with Strong Authentication

Server

Firewall

Client

sshd

port knock
client

Request sequence
Port Knocking with Strong Authentication

Challenge issued

Server

<table>
<thead>
<tr>
<th>sshd</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>•</td>
</tr>
<tr>
<td>•</td>
</tr>
<tr>
<td>•</td>
</tr>
</tbody>
</table>

Firewall

Client

port knock client
Port Knocking with Strong Authentication

Server

Firewall

Client

sshda

1

2

•

•

•

Response sequence:
HMAC-SHA1 (Key, Challenge)

port knock client

Rennie deGraaf, University of Calgary

ACSAC 2005
Port Knocking with Strong Authentication

Server

Firewall

Client

Port knock client

sshda

Response sequence:

HMAC-SHA1 (Key, Challenge)
Port Knocking with Strong Authentication

**Response sequence:**

HMAC-SHA1 (Key, Challenge)
Port Knocking with Strong Authentication

Response sequence:
HMAC-SHA1 (Key, Challenge)
Port Knocking with Strong Authentication

Response sequence:
HMAC-SHA1 (Key, Challenge)
Port Knocking with Strong Authentication

Server

Firewall

Client

sshd

1

2

•

•

•

22

ssh

Port is opened in the firewall
Packet Re-ordering

- Examined four methods
- Delay between sending
  - Slow
  - doesn’t allow packet loss detection
- Separate data and sequence number fields
  - Long sequences require either large port ranges or long execution times
Packet Re-ordering

- Encode data as a monotonically increasing sequence
  - Example: 1\textsuperscript{st} packet to [0, 255], 2\textsuperscript{nd} to [256, 511], 3\textsuperscript{rd} to [512, 767], etc.
  - Same run time as above, but requires fewer ports
  - Easier to use with disjoint port ranges
  - Optimal point: authentication in 0.278 seconds over 5120 ports on a slow network

Rennie deGraaf, University of Calgary
ACSAC 2005
Packet Re-ordering

Data:  73,  121,  92,  246,  149
Packet Re-ordering

Data:   73, 121, 92, 246, 149

Encode: send[i] = data[i] + 256*i + 1024

Send:   1097, 1401, 1628, 2038, 2197
Packet Re-ordering

Data: 73, 121, 92, 246, 149

Encode: \( \text{send}[i] = \text{data}[i] + 256 \times i + 1024 \)

Send: 1097, 1401, 1628, 2038, 2197

Recv: 2197, 1079, 1628, 1401, 2038

Rennie deGraaf, University of Calgary

ACSAC 2005
Packet Re-ordering

Data: 73, 121, 92, 246, 149

Encode: $send[i] = data[i] + 256 \times i + 1024$
Send: 1097, 1401, 1628, 2038, 2197
Recv: 2197, 1079, 1628, 1401, 2038

Decode: sort(recv)

$\text{data}[i] = \text{recv}[i] - 256 \times i - 1024$
Data: 73, 121, 92, 246, 149
Packet Re-ordering

- Send packets with sequence numbers congruent mod $n$ to the same range; others to different, unique ranges
  - Equivalent to previous method, except that the port range resets every $n$ packets
  - Example: $1^{\text{st}}$ packet to $[0, 255]$, $2^{\text{nd}}$ to $[256, 511]$, ..., $21^{\text{st}}$ to $[0, 255]$, ....
  - Chance of failure
  - Only useful for long sequences ($n > \sim 20$)
Weaknesses of Our Design

- No authentication-connection association
- If client is NATed, the server opens the port to the entire NATed network
- Knock sequences may be blocked by egress filters
- Failure on packet loss
Summary

- Port knocking is a practical way to add a light-weight authentication wrapper around existing services.
- Current port knocking implementations have a variety of problems.
- We have found solutions to several of these problems.
Questions?

Improved Port Knocking with Strong Authentication

Rennie deGraaf, John Aycock, and Michael Jacobson, Jr.

{degraaf,aycock,jacobs}@cpsc.ucalgary.ca

Department of Computer Science
University of Calgary
Calgary, Alberta, Canada