Advances In Single Packet Authorization

Michael Rash
Enterasys Networks, Inc.
http://www.cipherdyne.org/

ShmooCon
01/14/2006
Agenda

● Vulnerabilities vs. IDS/IPS
● Why another authentication / authorization method?
● Single Packet Authorization (SPA)
● Fwknop design and implementation
● New Features
● Disadvantages
● Future directions
● Live demo
Security Software Vulnerabilities

- Cisco IOS Firewall Authentication Proxy Buffer Overflow Vulnerability
- IPsec ESP Information Leak Vulnerability
- Check Point FW-1 Authentication Vulnerability
- OpenSSH GSSAPI Credential Disclosure Vulnerability
Cleartext IDS Over Encrypted Protocols

- WEB-MISC SSLv3 invalid timestamp attempt
- EXPLOIT SSLv2 Client_Hello with pad Challenge Length overflow attempt
- EXPLOIT gobbles SSH exploit attempt
- EXPLOIT ssh CRC32 overflow NOOP
- EXPLOIT ssh CRC32 overflow filler
Cleartext IDS Over Encrypted Protocols (cont'd)

print 'A'x1000;

(.)\1 {500}
Target Enumeration

# host www.yahoo.com
www.yahoo.akadns.net has address 216.109.117.206

# whois 216.109.117.206 | grep CIDR
CIDR: 216.109.112.0/20

# nmap -P0 -p T:22,256 -sS -sV -T Aggressive
216.109.112.0/20
Why Another Auth Method?

- Existing methods assume TCP/IP stack access
- Some application layer functions are available
- Strong crypto NOT enough
- Nmap
Goal: Minimize Available Code Paths

- Packet filters
- Stateful firewalls

```
# iptables -I INPUT 1 -j DROP
```
Main Question

Are DEFAULT DENY packet filters and simultaneous authenticated access compatible?
Answer: YES

- Authentication information passively collected (firewall logs, passive OS fingerprinting, netlink sockets, libpcap, libipq, etc.)
- Packet filter is dynamically reconfigured to allow temporary access
- Port Knocking
Single Packet Authorization

- Default deny stance for all protected services
- Packet filters reconfigured after SPA packet is received
- Uses passive monitoring strategy from the IDS world
- Encrypted, non-replayable, spoofable
- Any IP protocol can be used
- Up to minimum MTU number of bytes
Single Packet Authorization (cont')

- Integrates well with long-running protocols
- Adds authorization to previously unauthorized sessions
- Reduces false positive potential
- Nmap by itself cannot detect protected services (requires some packet to be generated in response to a scan).
- 0-day vulnerabilities more difficult to exploit
Single Packet Authorization vs. Port Knocking

- Both techniques use packet filters
- Both techniques passively collect information
- Replay attacks easily thwarted with SPA
- No port sequences to bust
- Much more data can be sent
- More difficult to detect (nothing to mistakenly detect as a port scan)
- Protocols without a notion of a “port” can be used
Disadvantages

- Additional key management
- Some services not readily compatible
- Session “piggy backing”
- Adds extra layer and associated time delay
- Authorization packets not transferred over reliable communication mechanism
- Not well suited to client protection
- libpcap vulnerabilities
Fwknop

- pcap, file_pcap, Netfilter pcap writer data collection methods
- Supports Rijndael and GnuPG
- Packets prepended with 16 bytes of random data
- Message integrity verified via internal MD5 sum
Fwknop (cont'd)

- Integrates with NAT
- Built-in spoofing capability (Net:::RawIP)
- Supports TCP, UDP, ICMP (default UDP/62201)
- Message replays stopped via MD5 sum cache
Fwknop (cont'd)

- Integrates with Netfilter policy via custom chains
- Supports access and command modes
New Features

- Supports multiple remote users and GPG signing keys
- OpenSSH-4.2p1 client integration
- Server side UNIX crypt() verification
- NAT Man-in-the-middle attacks prevented through automatic IP resolution via http://www.whatismyip.com/
- Client and server components separated (fwknop and fwknopd)
GPG Keys

[fwnkopd]$ gpg --gen-key
[fwnkopd]$ gpg -a --export <keyID> > server.asc
[fwnkopd]$ gpg --import client.asc
[fwnkopd]$ gpg --edit-key <clientKeyID>
Command> sign
SSH Usage

$ ssh -K "-A tcp/22 --gpg-recip ABCD1234 --gpg-sign 1234ABCD -w" user@host

GPG signing password:

  ->(netfilter reconfigured)<-

Password:

$ ssh -K "--last" user@host
Deployment Architectures

- Fwknop Client
- SPA Packet
- SSH
- Attacker 2
- Attacker 3
- Attacker 1
- Fwknopd Sniffer / Firewall
- Protected Network
- Dummy Target IP
Packet Format

Random data: 7808936091987532
Username: mbr
Timestamp: 1123247144
Version: 0.9.6
Action: 1 (access mode)
Access: 123.123.123.123,tcp/22
MD5 sum: y6tuSWoS+py7ppsESNR78A
<optional server authentication criteria>

7808936091987532:mbr:1123247144:0.9.6:0.0.0.0,tcp/22:y6tuSWoS+py7ppsESNR78A
Encrypted (Rijndael) Packets

udp/62201 (128 bytes):

Hul72UvwLqLqxIQLfTi7nXyjqIr37s8R9/JrYGcaP9PI4ADNK9pqeFghA20pXHwdpQf/TAbxt1L+GSwAkJBSP0USBRm6IK87+xBaVRpb9UNJ8HUw3DsRTXpcYXtqrPQP

ISTLpc2VMs2jGOJsJOAwIWxKChKUOMS88PttezX6u7TCsd7KVgzOIvjPRuSckjP/tbInEeMUK+53tKfvifNIX5vODinG5Cyi96XZThF2NO53dWN1dzQMV3dwPfbZdCab
Netfilter Integration

- Compatible with existing policy
- Custom fwknop chains (FWKNOP_INPUT)
- Most effective with connection tracking enabled
- Optional data collection via ULOG target
Example Netfilter Policy

Chain INPUT (policy DROP)

FWK NOP_INPUT all -- 0.0.0.0/0 0.0.0.0/0
ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
ACCEPT tcp -- 192.168.10.3 0.0.0.0/0 tcp dpt:80
ULOG udp -- 0.0.0.0/0 0.0.0.0/0 udp dpt:62201 ULOG
copy_range 0 nlgroup 1 prefix `FWK NOP' queue_threshold 1

Chain FWK NOP_INPUT (1 references)

ACCEPT tcp -- * * 192.168.10.2 0.0.0.0/0 tcp dpt:22
/etc/fwknop/fwknop.conf

EMAIL_ADDRESSES                    mbr@cipherdyne.org;
AUTH_MODE                               PCAP;
PCAP_INTF                                  eth1;
ENABLE_PCAP_PROMISC           Y;
PCAP_FILTER                               udp port 62201;
PCAP_PKT_FILE                           /var/log/ulogd.pcap;
ENABLE_MD5_PERSISTENCE     Y;
/etc/fwknop/access.conf

SOURCE: ANY;
DATA_COLLECT_MODE: PCAP;
OPEN_PORTS: tcp/22;
PERMIT_CLIENT_PORTS: Y;
#ENABLE_CMD_EXEC: Y;
KEY: <encryptkey>;
GPG_DECRYPT_ID: ABCD1234;
GPG_DECRYPT_PW: <password>;
GPG_REMOTE_ID: 1234ABCD;
FW_ACCESS_TIMEOUT: 10;
REQUIRE_USERNAME: mbr;
IDS Alert Reduction

- Most IDS's are stateful
- Sessions can only be established after authorization
- Less probability of arbitrary malicious sessions
Future Directions

- Add support for additional authentication infrastructures (LDAP, Kerberos, Radius, etc.)
- Additional client integration (VPN clients, Web browsers)
- GUI development
- Potential kernel stack extensions (NDIS driver on Windows, IP stack patch for Linux)
Live demo...
Questions?

http://www.cipherdyne.org/fwknop/

mbr@cipherdyne.org