Generalized Single Packet Authorization for Cloud Computing Environments

Michael Rash

http://www.cipherdyne.org/

ShmooCon

February, 2013
Agenda

- Brief Port Knocking / Single Packet Authorization Primer
- Lengthy demo – SPA integrated into Amazon's Cloud
- General integration points for Cloud providers
- fwknop-2.5 release – HMAC-SHA256 support
- Where is SPA headed?
PK/SPA Assertion

There is a security benefit in service concealment behind a default-drop packet filter + plus a lightweight passive authentication layer

(Not a defense for client-side vulnerabilities)
No Shortage of Server Vulns

- Cisco rsh vuln (HD Moore: “Hacking Like It's 1985”):
  http://goo.gl/gL6ZJ (https://community.rapid7.com/community/metasploit/blog...)
- UPnP vulnerabilities (affecting millions of devices):
  https://community.rapid7.com/docs/DOC-2150
- SHODAN enumeration of Internet connected SCADA devices:
- Barracuda Networks SSH backdoors (Stefan Viehböck):
  http://krebsonsecurity.com/2013/01/backdoors-found-in-barracuda-networks-gear/
Typical PK/SPA Work Flow

- User wants SSH access behind PK/SPA firewall
- User executes PK/SPA client
- Firewall is reconfigured to allow SSH connections from the specified IP
- PK/SPA packet(s) passively monitored
- PK/SPA packet(s) never acknowledged in any way
- SSHD cannot be scanned for
- *Think beyond SSHD*
General Goal of fwknop

Solve PK limitations while simultaneously retaining its benefits
The fwknop Design

- Firewall default drop stance for protected services
- Passive collection of authentication information (libpcap*)
- Support for Symmetric and Asymmetric ciphers
- Encrypted and non-replayable SPA packets
  - Do not want anything that trusts an IP in the network layer header
- Server portable to embedded systems
  - Do not want a heavyweight interpreted language (this is a trade off)
- Server portable to different firewall architectures and router ACL languages
  - Make sophisticated use of NAT
- Client portable to everything from Cygwin to the iPhone
  - Do not want to require raw socket manipulation of packet headers or admin privileges
- Minimize library dependencies
Things Aren't Always as They Seem

- User gains access to NetB from NetA with SPA
- Attacker: Which system to attack?
- SPA server can be anywhere on the routing path of an SPA packet – not just the SPA destination IP
- SPA packet source IP can be spoofed too
- *Neither the SPA source nor destination IP matters*
Tutorial


http://www.cipherdyne.org/fwknop/docs/fwknop-tutorial.html
SPA in the Amazon Cloud

http://aws.amazon.com/
“...4.2 Other Security and Backup. You are responsible for properly configuring and using the Service Offerings and taking your own steps to maintain appropriate security, protection and backup of Your Content, which may include the use of encryption technology to protect Your Content from unauthorized access and routine archiving Your Content...”

http://aws.amazon.com/agreement/
The Perfect SPA Use Case

• Microsoft RDP vulnerability last year (CVE-2012-0002)
• Full remote code execution potential, although Metasploit only has a DoS module
• For a time, Cloud provider Windows images were vulnerable

• Problem: fwknop does not support a Windows firewall
Amazon VPC + SPA Setup

Amazon ACL only allows web connections from arbitrary IP's (and UDP port 40001 for SPA packets in this case). The scanner can ONLY ever see the Ubuntu webserver.
**fwknopd Configuration**

- We're going to create an SPA “jump” host gateway

```bash
# cat /etc/fwknop/fwknopd.conf
PCAP_FILTER               udp port 40001;
ENABLE_IPT_FORWARDING     Y;
ENABLE_IPT_LOCAL_NAT      Y;
ENABLE_IPT_SNAT           Y;
SNAT_TRANSLATE_IP         10.0.0.12;

# cat /etc/fwknop/access.conf
SOURCE: ANY;
KEY: test1234;
FORCE_NAT: 10.0.0.12 22;
REQUIRE_SOURCE: Y;

SOURCE: ANY;
KEY: 1234test;
REQUIRE_SOURCE: Y;
```
Demo (video)
Demo: Key Points

- Do not have any direct integration with AWS border controls
- All SPA principles apply
  - Default-drop firewall policy – cannot scan for a target
  - Passive packet acquisition – SPA packets are never acknowledged
  - Replay detection
  - Temporary firewall reconfiguration for service access
- Access to any service on any VPC system all through a single routable Elastic IP
  - SPA hardened “jump” host
  - Sophisticated usage of NAT
  - Accessed hosts don't even need a route to the Internet (DNAT + SNAT usage)
- “Ghost” services
  - Scanners only see Apache (or whatever), but SPA allows access to SSHD or any other service
  - iptables SPA NAT rules intercept connections out from under local userspace services
  - fwknop has supported ghost services since the old perl days
Can We Generalize This to Other Cloud Computing Environments?
Some Observations About Amazon

- Could fully control and configure internal OS images (install software, manipulate firewall rules, etc.)
- No (apparent) specialized filtering in AWS border ACL
- Not restricted to accessing VPC hosts with specialized applications controlled by Amazon – any application that is compatible with ACL configuration will work
- The above translates to greater ease of use and deployment for Amazon customers independent of SPA or anything else – e.g. it is a good architecture that other Cloud providers will emulate
SPA Integration with Arbitrary Cloud IaaS Providers

- “Useful” Cloud infrastructures provide remote access via SSH/RDP/VPN protocol to customizable OS images
  - Universal HTTP/HTTPS for Cloud usage is not generally compatible with SPA
- Cloud providers usually implement a network ACL capability
  - May or may not be customizable by the user
  - SPA client must communicate in a compatible fashion
- We don't necessarily need NAT capabilities in the SPA implementation (support less complex cloud environments)

- IaaS (Infrastructure as a Service) providers are generally SPA-compatible
Cloud Providers

• Wikipedia currently lists 129 different Cloud providers:

http://en.wikipedia.org/wiki/Category:Cloud_computing_providers
Private Clouds

- Bare metal owned by a private entity
- Cloud layer provided by open source or proprietary computing stack
- SPA is likely compatible in two ways:
  - Integration with raw OS underneath the virtualization layer
  - Integration with guest OS instances (e.g. similar to AWS deployment)
Hybrid Clouds

- SPA is likely compatible *bi-directionally* if public portion is compatible
Evaluating Cloud IaaS + SPA Compatibility

- Can run full OS images?
  - Yes
    - Allow SSH/RDP/VPN?
      - Yes
        - ACL compatible with SPA?
          - Yes
            - Full SPA Compatibility +/- NAT to other internal Cloud hosts
          - No
            - Test SPA over allowed proto/port (UDP/53?)
            - Yes
            - No
              - Test SSH/RDP/VPN over allowed proto/port (TCP/80?)
              - Yes
              - No
                - Complex integration effort (Cloud provider)

  - No
    - N

Copyright (C) Michael Rash 2013
Moving Up the Cloud Stack

- We've shown SPA integrates well with IaaS, but what about PaaS (Platform as a Service) and SaaS (Software as a Service) models?
Moving Up the Cloud Stack (cont'd)

- SPA PaaS integration to the extent that the base infrastructure is under user control
  - Amazon Elastic Beanstalk
- SaaS not generally SPA compatible
  - Users do not have infrastructure control
  - Would require massive integration effort, and drastically changes usage model
Amazon Elastic Beanstalk


```
<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EC2 Instance Type</strong></td>
<td>t1.micro</td>
</tr>
<tr>
<td><strong>EC2 Security Groups</strong></td>
<td>elasticbeanstalk-default</td>
</tr>
<tr>
<td><strong>Existing Key Pair</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Monitoring Interval</strong></td>
<td>5 minute</td>
</tr>
<tr>
<td><strong>Custom AMI ID</strong></td>
<td>ami-5e4db237</td>
</tr>
</tbody>
</table>

*Note: It may take a few minutes to see changes to these options take effect in your environment.*
Specialized Cloud Providers

- Cloud storage providers (DropBox, Mozy, etc.)
  - Not generally SPA-compatible (SaaS model)
  - Such providers construct purpose-built cloud infrastructure that is accessed through a dedicated client-side application (web browser or custom app)

- Clouds optimized for computing performance (e.g. Penguin Computing)
  - SPA compatibility likely for IaaS portion
  - SPA not generally a good fit for HPC jobs
Further Research...

- To what extent are packet filters used within Cloud computing stacks? (Independent of OS packet filters.)
  - This may hint at direct SPA integration with Cloud software
- Are there natural SPA integration points for distributed computing jobs?
  - If so, is there a security benefit? (Cloud-specific threat modeling.)
  - Are there integration points for admin layers below distributed content distribution services (e.g. Amazon Cloud Front)?
- Do any major IaaS Cloud providers leverage packet filters in ways that are incompatible with SPA? (Probably not.)
fwknop Development
fwknop-2.5 (coming soon)

• HMAC-SHA256
  - HMAC(K, m) = H((K ⊕ opad) || H((K ⊕ ipad) || m))
  - SPA encrypted message = m || HMAC
  - K != encryption key

• fwknop uses the encrypt-then-authenticate paradigm
  - SSH uses encrypt-and-MAC
  - SSL uses MAC-then-encrypt ← Has made the Vaudenay and more recent “Lucky 13” padding oracle attacks possible
  - IPSEC uses encrypt-then-MAC ← INT-CTXT and IND-CCA2 secure
fwknop Vulnerabilities

• CVE-2012-4435 — Improper IP validation (requires a valid encryption key to exploit)

• CVE-2012-4436 — Client side --last processing overflow (local exploit)

• Fixed since 2.0.3. (Latest release is 2.0.4)

• CREDIT: Fernando Arnaboldi, IOActive. Additional thanks to Erik Gomez for helping to make this auditing effort possible.
What are we doing about this?

- Test suite driven valgrind validation
  - Every new commit is tested against a valgrind baseline
  - Lightweight C code helps a lot here
- SPA packet fuzzer
- Compile time security options
- Usage of static analyzers (e.g. splint, Clang static analyzer, etc.)
- SPA protocol review
SPA Packet Fuzzer

- Builds encrypted SPA packets with malicious payloads
- Series of patches against libfko to remove various constraints and validation steps
- Automatically tested via the `test/test-fwknop.pl` test suite
- Over 2,000 fuzzing packets currently used in different modes
Test Suite:

# ./test-fwknop.pl

[build security] [client] Position Independent Executable (PIE).....pass (3)
[build security] [client] stack protected binary......................pass (4)
[build security] [client] fortify source functions.....................pass (5)
[build security] [client] read-only relocations.......................pass (6)
[build security] [client] immediate binding........................pass (7)
[build security] [server] Position Independent Executable (PIE).....pass (8)
[build security] [server] stack protected binary......................pass (9)
[build security] [server] fortify source functions.....................pass (10)
[build security] [server] read-only relocations.......................pass (11)
[build security] [server] immediate binding........................pass (12)

- This is enabled via:
  - gcc ... -fstack-protector-all -fstack-protector -fPIE -pie -D_FORTIFY_SOURCE=2
    -Wl,-z,relro -Wl,-z,now
iPhone + Android fwknop Clients
The Future of fwknop

- Mandatory Access Control support via SELinux and/or AppArmor
- Further cloud computing extensions and integration points
- Privilege separation
- Support for libcap-ng
- UDP listener mode
- Tunneling mode extensions (DNS, HTTP, SMTP, Tor)
Linux Firewalls 2nd Edition
To be released in 2014...
Thank You...

• The Amazon Security team
• Damien Stuart – developed the original C port
• Fernando Arnaboldi and Erik Gomez (IOActive)
• Franck Joncourt (Debian)
• Sebastien Jeanquier – authoritative PK/SPA thesis
• Sean Greven (FreeBSD port)
• Vlad Glagolev (OpenBSD port)
Questions?

mbr@cipherdyne.org

@michaelrash

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

Slides:

http://www.cipherdyne.org/talks/ShmooCon_2013_mrash_Cloud_SPA.pdf
http://www.cipherdyne.org/talks/ShmooCon_2013_mrash_Cloud_SPA_demo.mpg4