Open Source Security Orchestration

Brucon 9, Ghent 2017



Hellfire Security

Gregory Pickett, CISSP, GCIA, GPEN

Chicago, Illinois

gregory.pickett@hellfiresecurity.com

Overview

- How This All Began
- Orchestrating All The Things
- Behold Skynet
- Making It Better
- Wrapping Up

Original Question

- Multiple Cloud Servers
- All Using Fail2Ban to Protect Themselves
- Can I share Fail2Ban jails between these Servers?

Other Questions

- How do we get to threats in time?
- How do we make sure that the evidence gets captured?
- How do we make sure that the threat is stopped before it is too late?
- How do we do this with a limited staff?

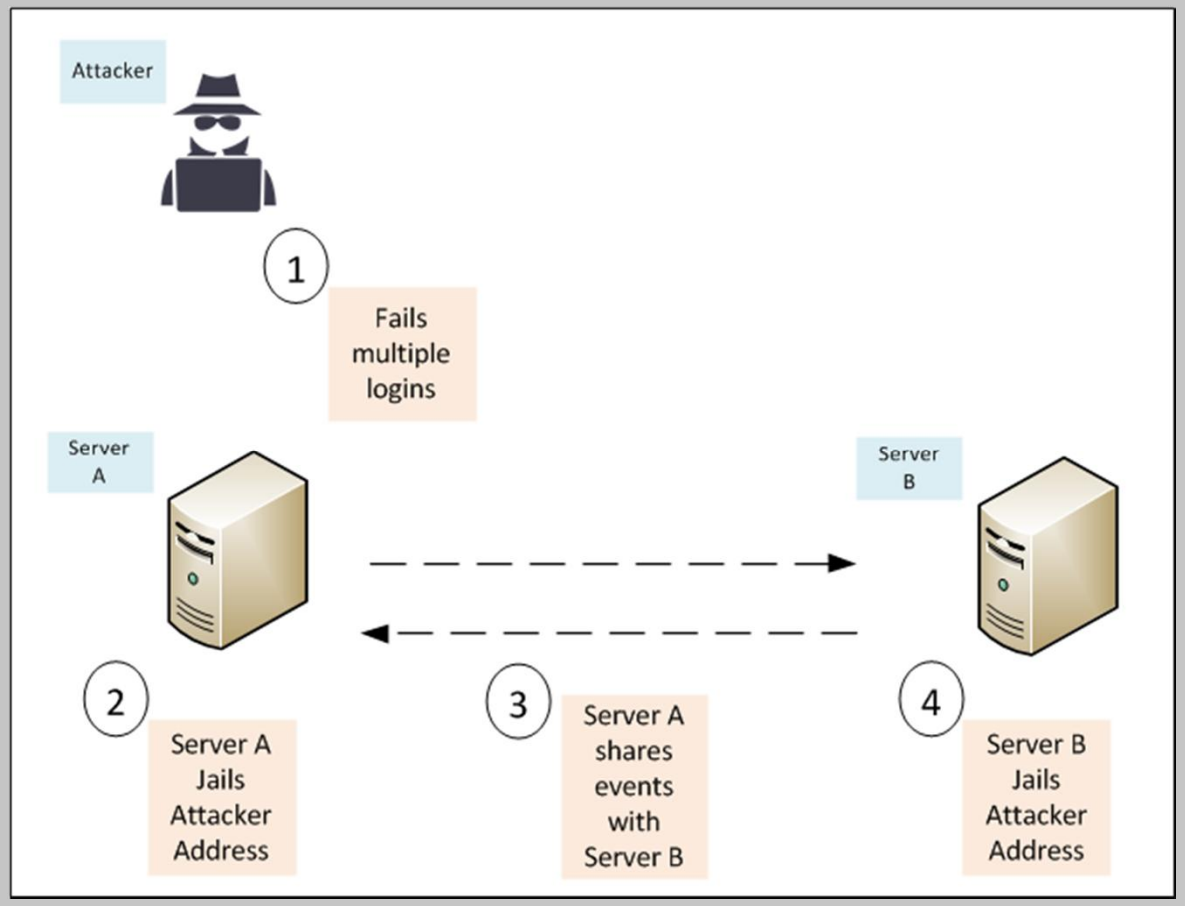
This Is Because

- Security Operations
  - Monitor The Enterprise
  - Process Alerts (or Correlations)
  - Kick Off Incident Response
- Despite Multitude of Solutions
  - Still A Manual Process!
  - Each Solution Kicked Off In Sequence By Us
- A Lot of Time Is Wasted Being A Bridge Between Systems

What I Want

- Keep Doing What Your Doing
- Talk Directly To Each Other
- Get What You Need from Each Other
- Leave Me Out Of It

How This Would Work



Use Cases



Generate Threat Intelligence Feed

- Received Events From Peers
- Generate A Blacklist from Source of Threat Events
- Use With Anything That Can Consume A Blacklist
  - Firewalls
  - Endpoint Solutions
  - Detection Tools
- Share The Blacklist with Vendors, Partners, and Colleagues

Firewall Rule Propagation

- Receives Events From Peers
  - Host Firewall
  - Network Firewall
- Blocks Source of Threat Events
- Distributes Events Among Peers
  - Host Firewall
  - Network Firewall

Drop Propogation

- Drop Source of Threat Events
- Distributes Events Among Peers
  - Web Application Firewalls
  - Intrusion Prevention Systems

Prevent Known Threats

- Receives Events From External Threat Feeds
  - Host Firewall
  - Network Firewall
- Blocks Source of Threat Events

NAT to Honeypot

- Receives Events From Peers
  - Host Firewall
  - Network Firewall
- Redirects Source of Threat Away From Assets

NAT to Tarpit

- Receives Events From Peers
  - Host Firewall
  - Network Firewall
- Slows Down Source of Threat

Capture Threat Activity

- Receives Events From Peers
  - Switches
  - Routers
  - Firewalls
- Runs Packet Capture on Source of Threat Activity

Inject Beacon

- Receives Events From Peers
  - FTP Server
  - File Servers
  - Honey Pots
- Drops Beacon into Path of Source of Threat Activity

Redirect Traffic

- Receives Events From Peers
  - Routers
  - Firewalls
- Changes the Route for Source of Threat Activity
  - Run Their Traffic Through Different Segment
  - Segment Contains Additional Inline Sensors
  - Afterwards, It Proceeds to Destination

Reporting Threats

- Receives Events From Peers
  - Email Server
- Reports Source of Threat to Abuse Address

Host Isolation

- Receives Events From Peers
  - Switches
  - Routers
  - Firewalls
- Applies ACL to Target of Threat Activity

Additional Logging

- Receives Events From Peers
  - Switch
  - Router
  - Firewall
  - Server
  - Application
- Verbose Logging for Source of Threat Activity
- Verbose Logging for Target of Threat Activity

Trigger Password Resets

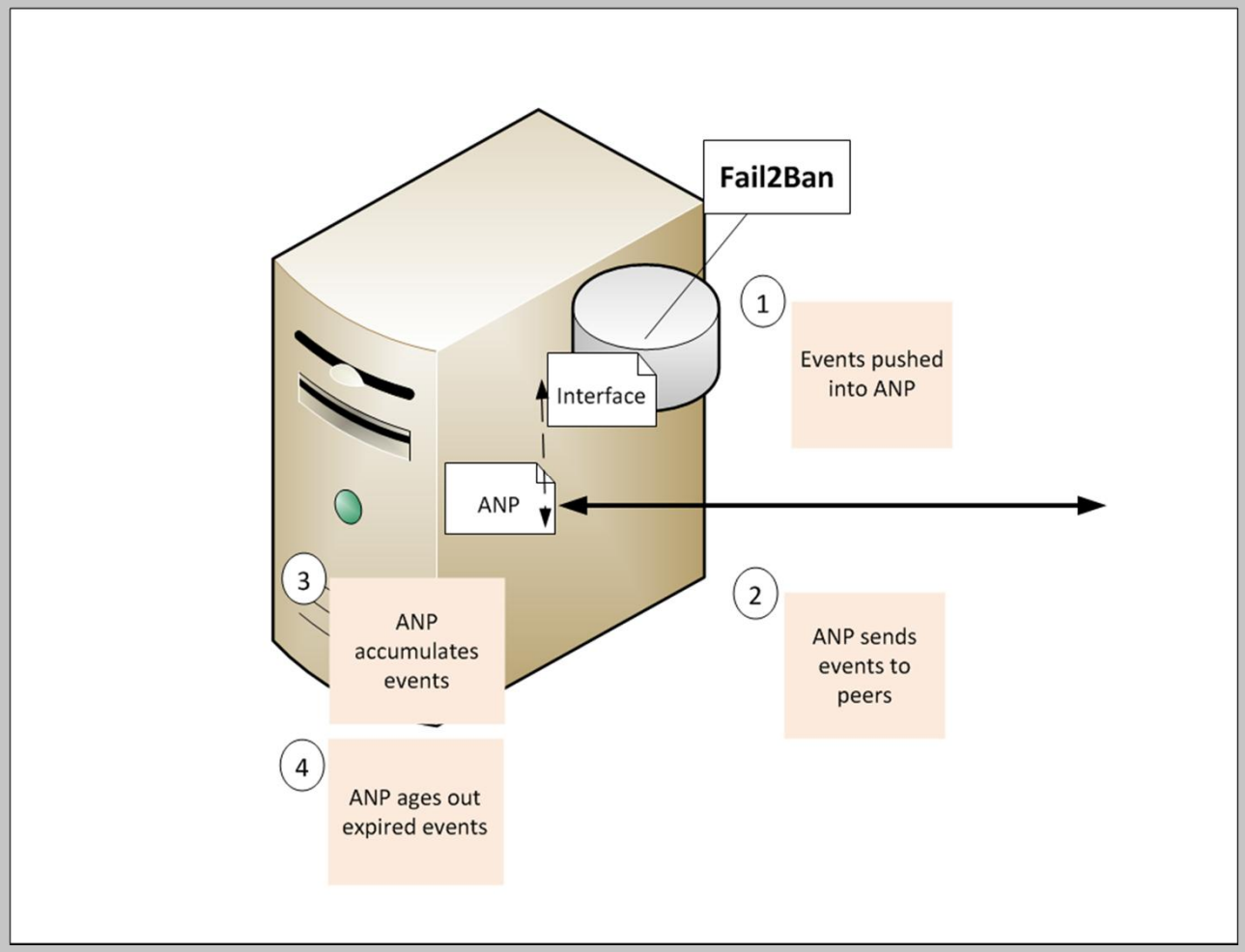
- Receives Events From Peers
  - LDAP
  - Active Directory
  - Radius
  - TACACS+
- Starts Password Reset Process for Target of Threat

Security Orchestration

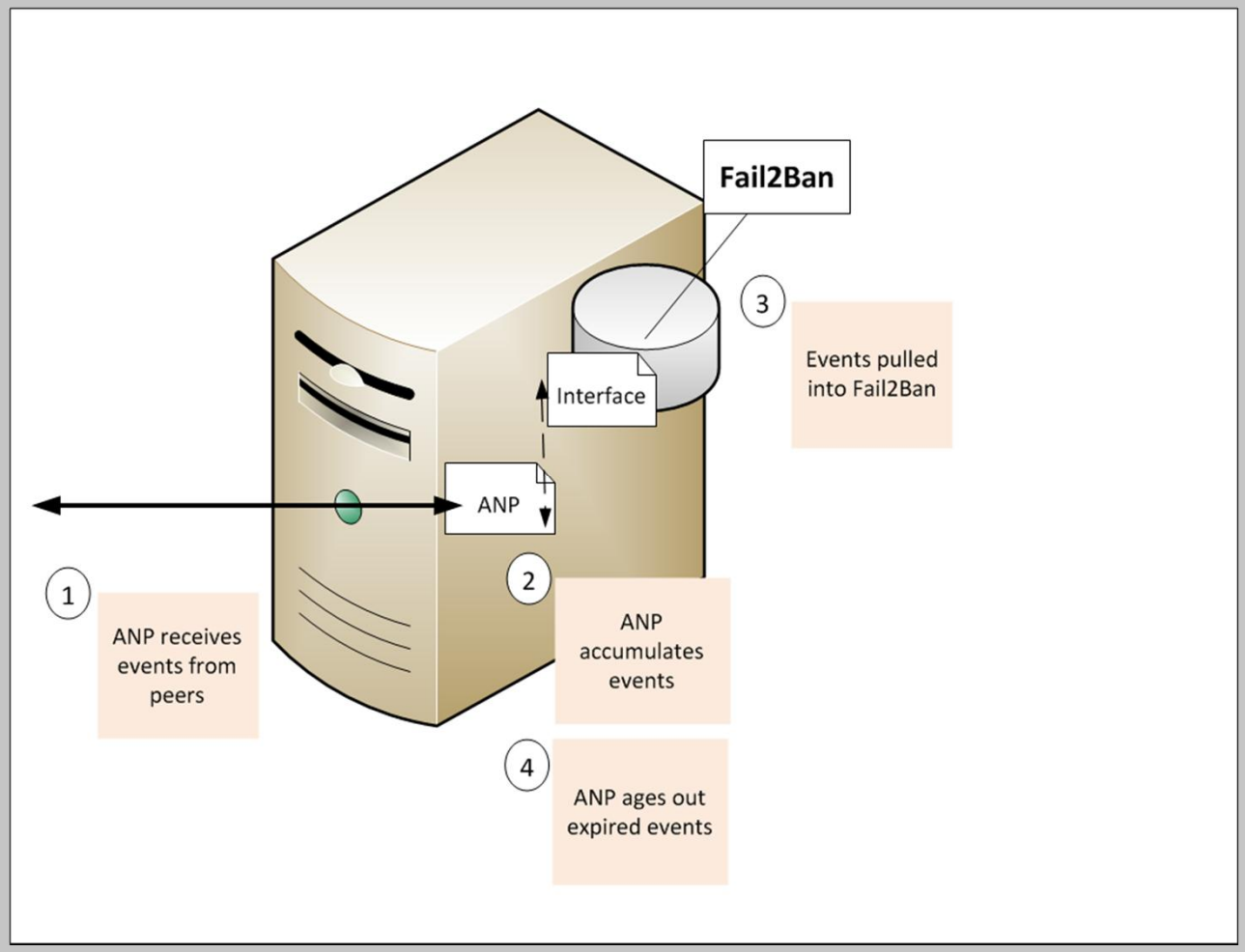
Adaptive Network Protocol (ANP)

- Shares Events Between Systems In Common Format
- Events Are Stored Locally
- Peers Make Use of Shared Events How They See Fit
  - fail2ban
  - modsecurity
  - ipTables

Server A


Fail2Ban
1
Events pushed into ANP
Interface
ANP
2
ANP sends events to peers
3
ANP accumulates events
4
ANP ages out expired events

Server B


Fail2Ban
3
Events pulled into Fail2Ban
Interface
ANP
1
ANP receives events from peers
2
ANP accumulates events
4
ANP ages out expired events

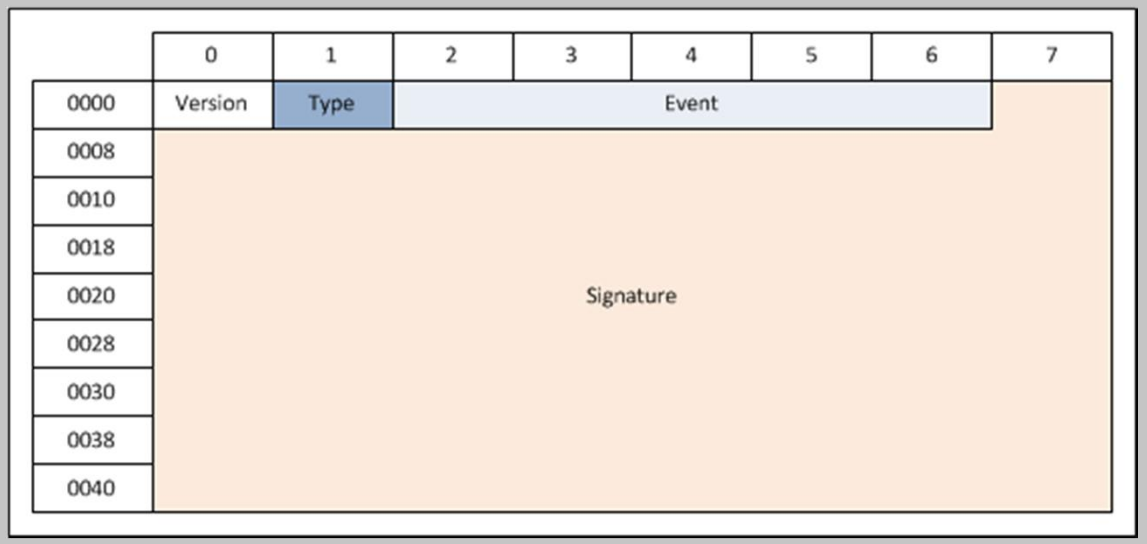
Protocol

- Sharing
  - Multicast to Local Peers
  - Unicast to Remote Peers
- Messages
  - Add Threat Event
  - Remove Threat Event

Protocol

- Operations
  - Sends and Receives from local peers on UDP Port 15000
  - Receives from remote peers on TCP Port 15000
  - Every message signed with SHA256
- Rules
  - The Signature Must Be A Good Signature
  - If Already Known, Do Not Share
  - Do Not Reflect Back To The Source

Packet



- Version is 1 Byte
- Type is 1 Byte
- Event is Variable
- Signature is 64 Bytes

Packet

```
▷ Frame 1: 113 bytes on wire (904 bits), 113 bytes captured (904 bits)
▷ Ethernet II, Src: VutlanSr_52:86:2f (00:23:98:52:86:2f), Dst: IPv4mcast_01 (01:00:5e:00:00:01)
▷ Internet Protocol Version 4, Src: 192.168.2.101, Dst: 224.0.0.1
▷ User Datagram Protocol, Src Port: 63090, Dst Port: 15000
◢ Data (71 bytes)
    Data: 01010808080801aba39920e5fb518b37fde4510ec01f4bb4...
    [Length: 71]
```

```
0000  01 00 5e 00 00 01 00 23  98 52 86 2f 08 00 45 00   ..^....# .R./..E.
0010  00 63 5a f9 00 00 01 11  bb 82 c0 a8 02 65 e0 00   .cZ..... .....e..
0020  00 01 f6 72 3a 98 00 4f  fc ed 01 01 08 08 08 08   ...r:..O ........
0030  01 ab a3 99 20 e5 fb 51  8b 37 fd e4 51 0e c0 1f   .... ..Q .7..Q...
0040  4b b4 6e 0e a9 4d bd b9  68 40 35 5d dd 47 67 16   K.n..M.. h@5].Gg.
0050  32 b4 39 0c 5e ff 66 dd  5f d8 2c 0a 03 af 57 e3   2.9.^.f. _.,...W.
0060  12 26 ff b2 d7 0c 42 ad  ce 8e d0 25 59 1f b1 30   .&....B. ...%Y..0
0070  d1                                                .
```

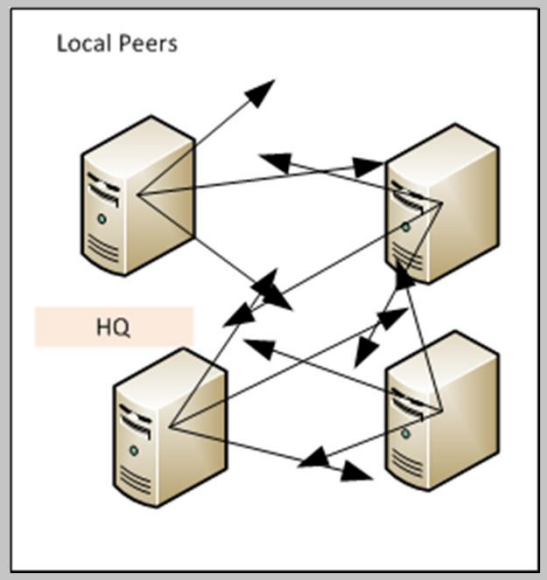
Messages

- Add Threat Event
  - Address
  - Time-To-Live (TTL)
- Remove Threat Event
  - Address
  - Time-To-Live (TTL)

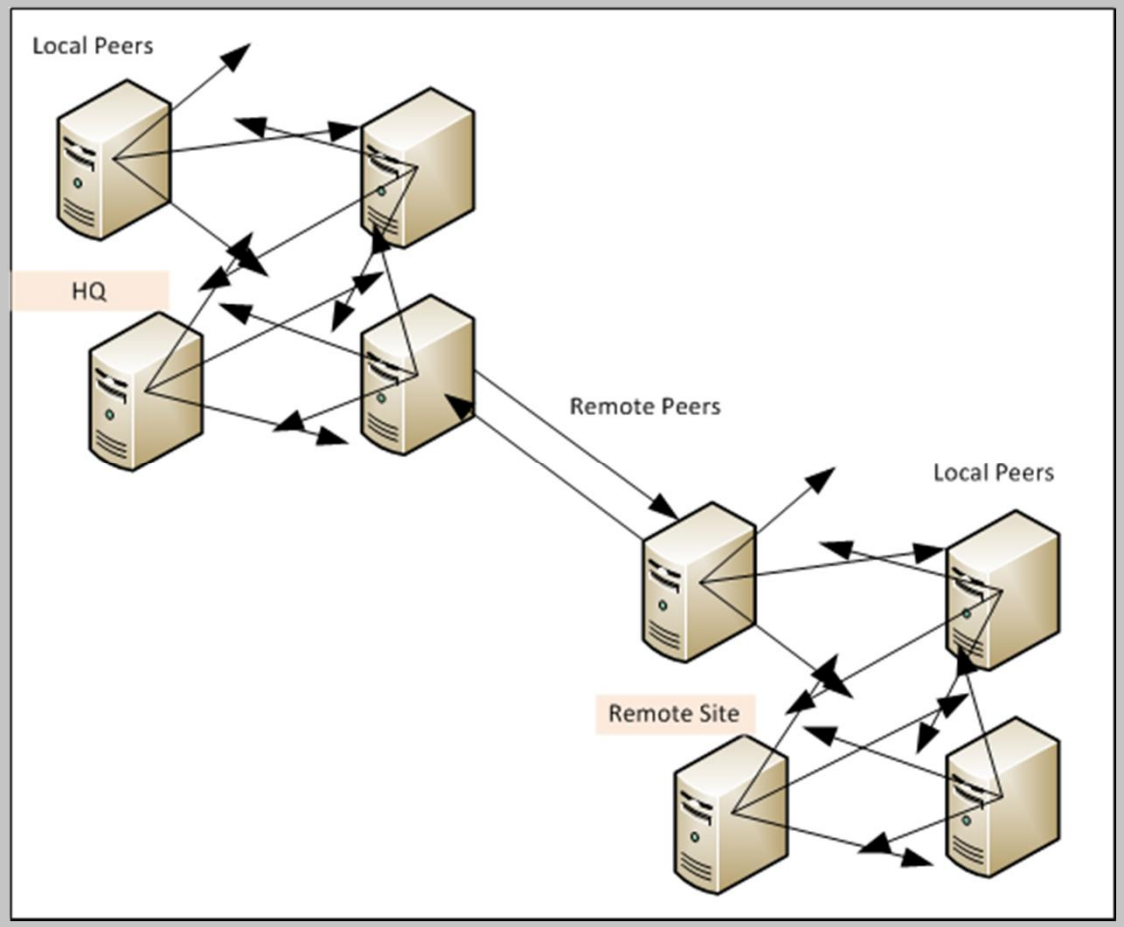
Peering

- Local
- Remote
  - Same Network
    - Across Same Location
    - Across Different Locations
    - Link-up Cloud Resources
  - Different Networks

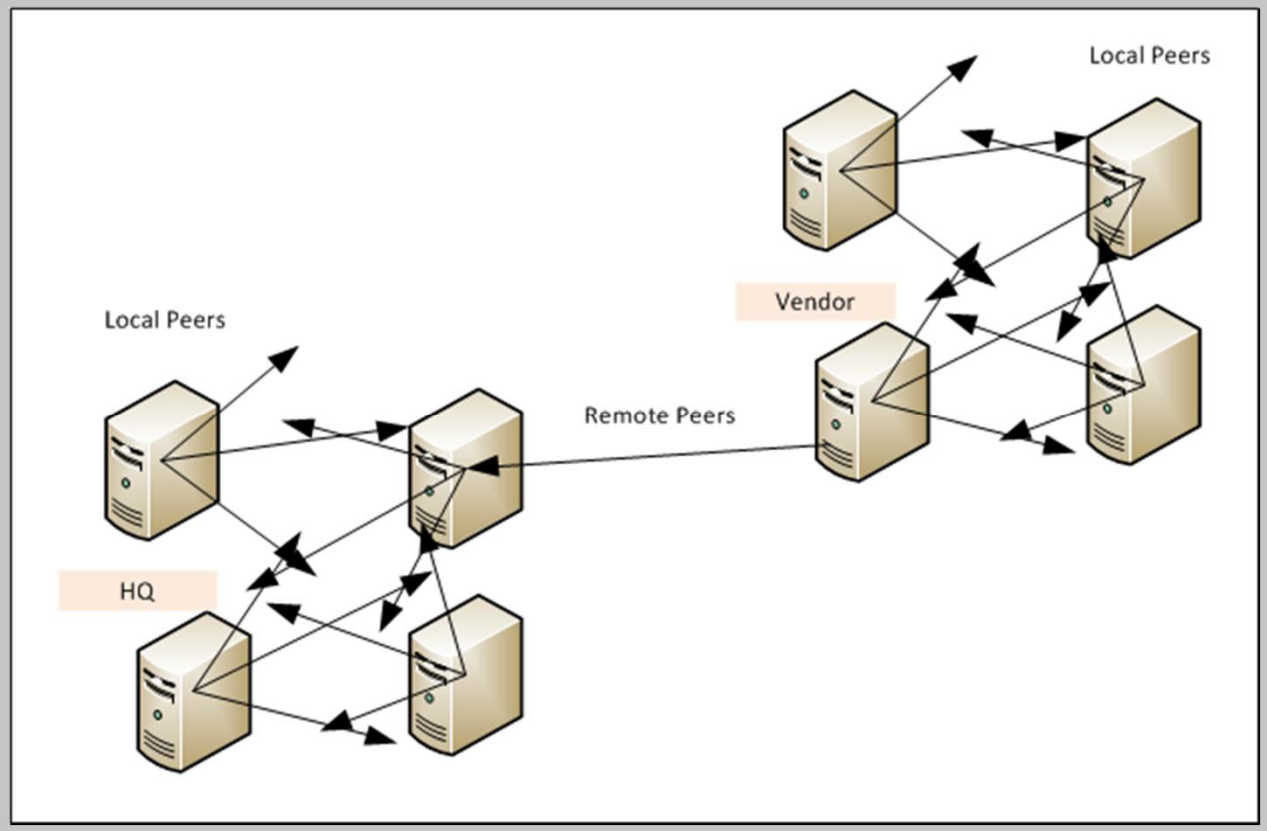
Single Location



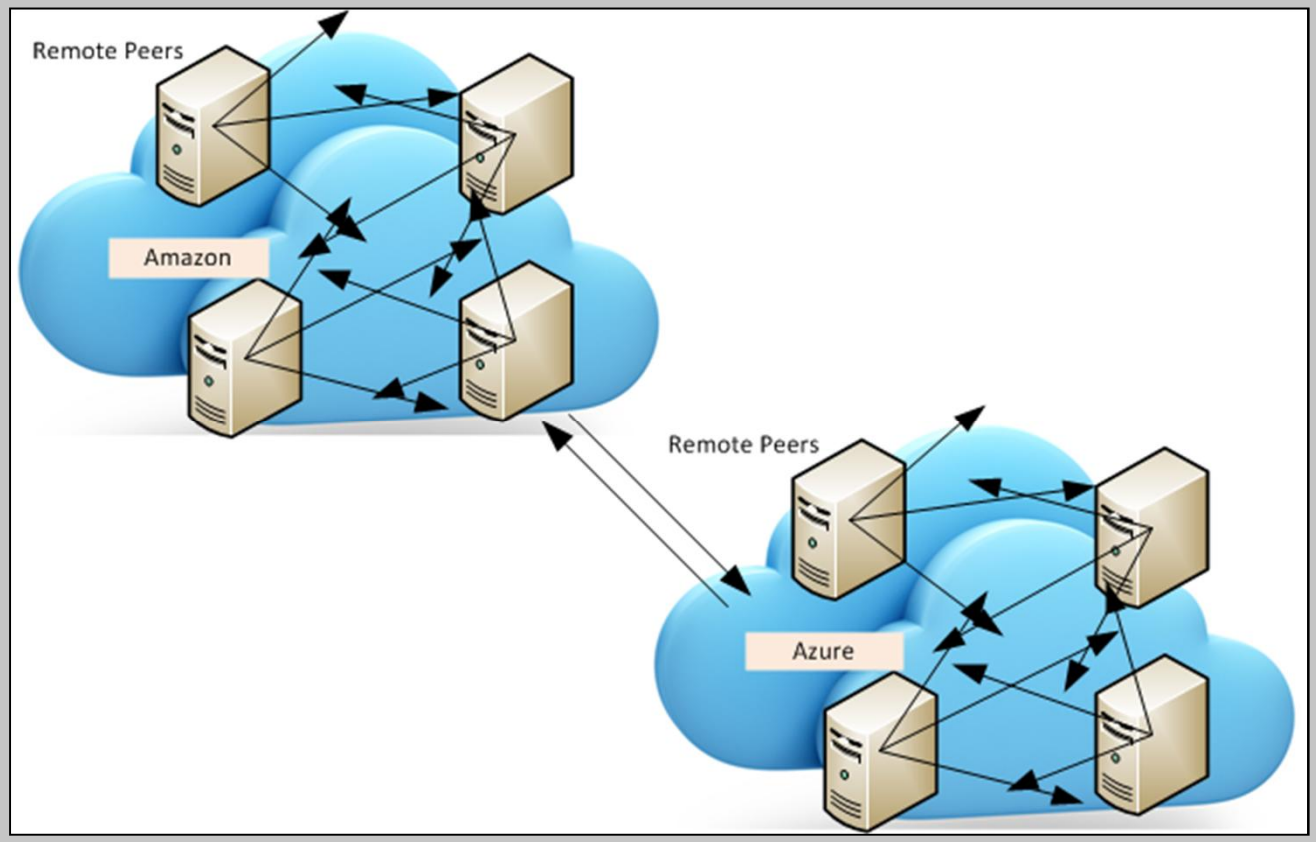
Multiple Locations



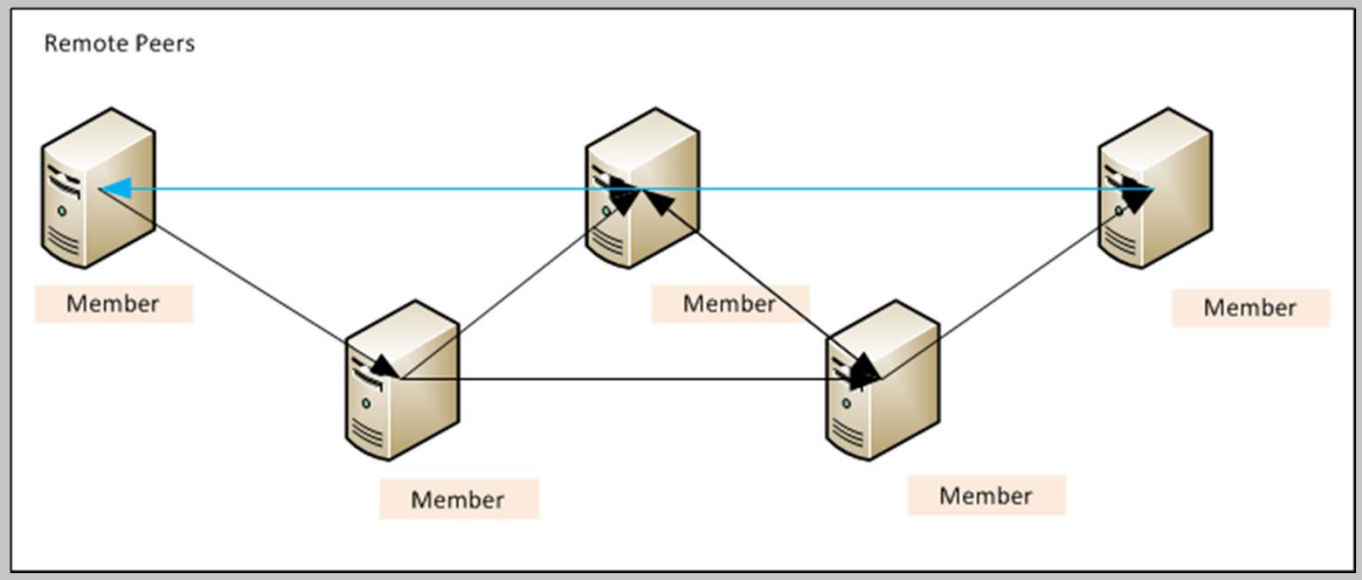
Trusted Partner or Vendor



Cloud Assets



Communities

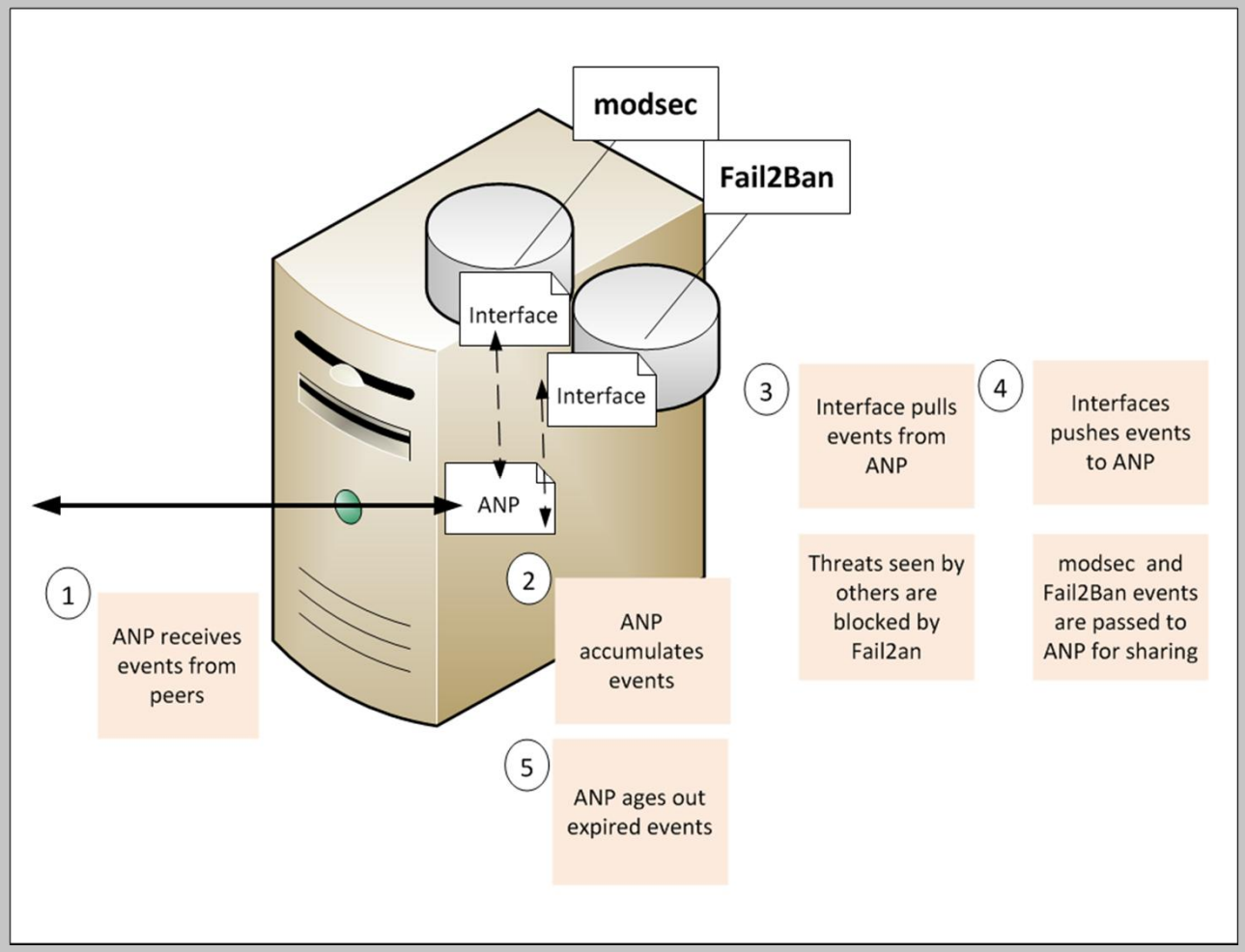


Interfaces

What They Do

- Purpose
  - Publish Events to ANP
  - Pull Events From ANP
- Components
  - Supporting
  - Writer
  - Reader
- Operations
  - Publishes via Loopback interface
  - Pulls from via published lists

What They Do


modsec
Fail2Ban
Interface
Interface
ANP
1
ANP receives events from peers
2
ANP accumulates events
3
Interface pulls events from ANP
4
Interfaces pushes events to ANP
Threats seen by others are blocked by Fail2an
modsec and Fail2Ban events are passed to ANP for sharing
5
ANP ages out expired events

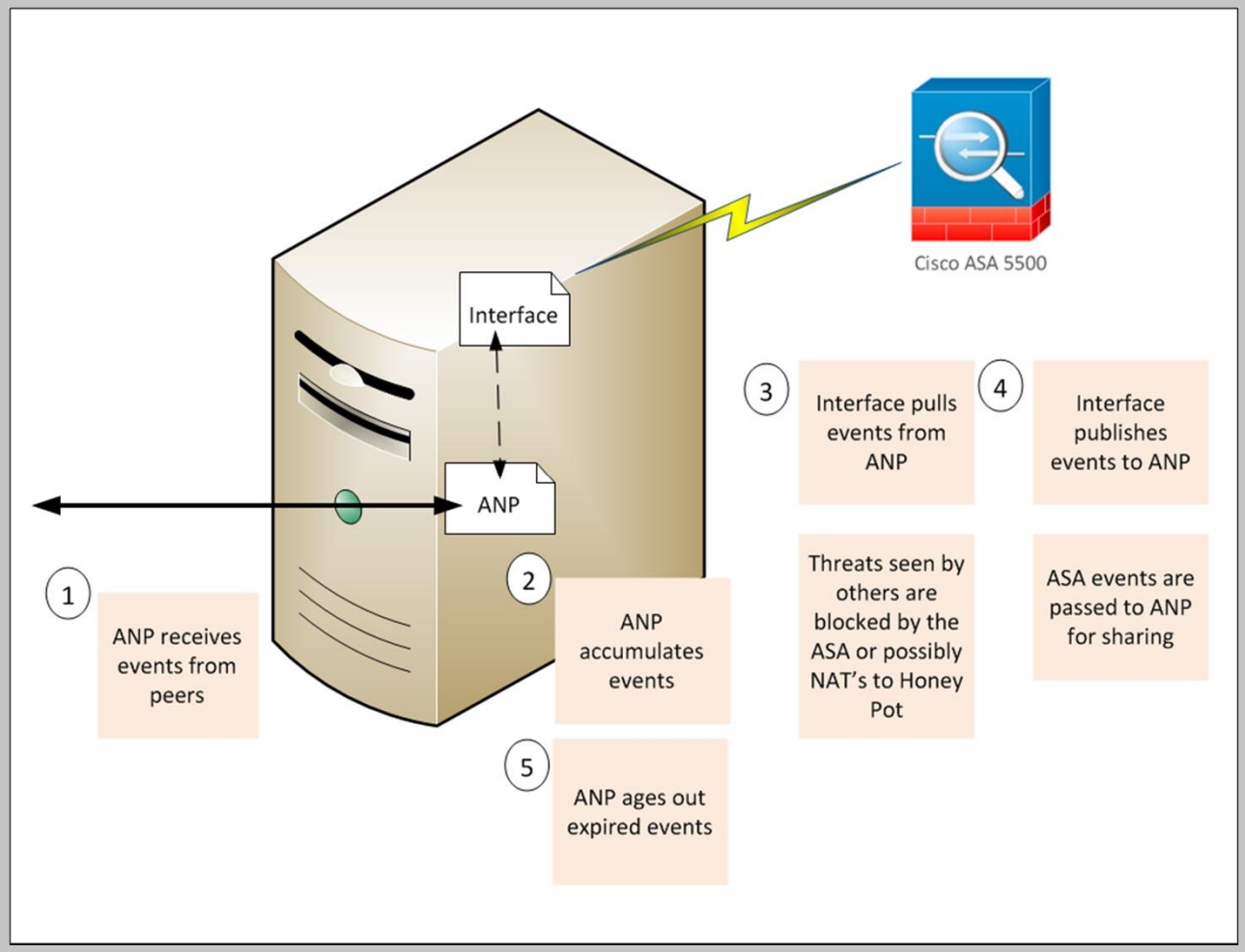
Native

- Integrated Solution
  - ANP installed on the same system
  - Read and Writes Locally
- Examples
  - Fail2Ban
  - Iptables
  - modsec

Surrogate

- Stand Alone Solution
  - ANP installed on a different system
  - Read and Writes to the Remote (Stand Alone) Solution
- Examples
  - ASA
  - Switch
  - Router

Surrogate


Cisco ASA 5500
Interface
ANP
1
ANP receives events from peers
2
ANP accumulates events
3
Interface pulls events from ANP
4
Interface publishes events to ANP
Threats seen by others are blocked by the ASA or possibly NAT's to Honey Pot
ASA events are passed to ANP for sharing
5
ANP ages out expired events

Existing Interfaces

Fail2Ban

- Pulls Events
  - Reads Threat Events from ANP
  - Adds Threats to Jail
- Publishes Events
  - Writes Jailed Addresses to ANP
- Because of ANP Aging, this means threats stay jailed for 24 hours
- Mistakes can be reversed using an additional tool to inject a Remove Threat event

Blacklist

- Pulls Events
  - Reads Threat Events from ANP
  - Adds Threats to Blacklist
- Distribute for Internal or External Use
  - Detecting
  - Blocking
  - Threat Indicator

modsec

- Publishes Its Events
  - Writes Attacker Addresses to ANP
- Pair with IPTables interface
- NAT attackers to Honeypot

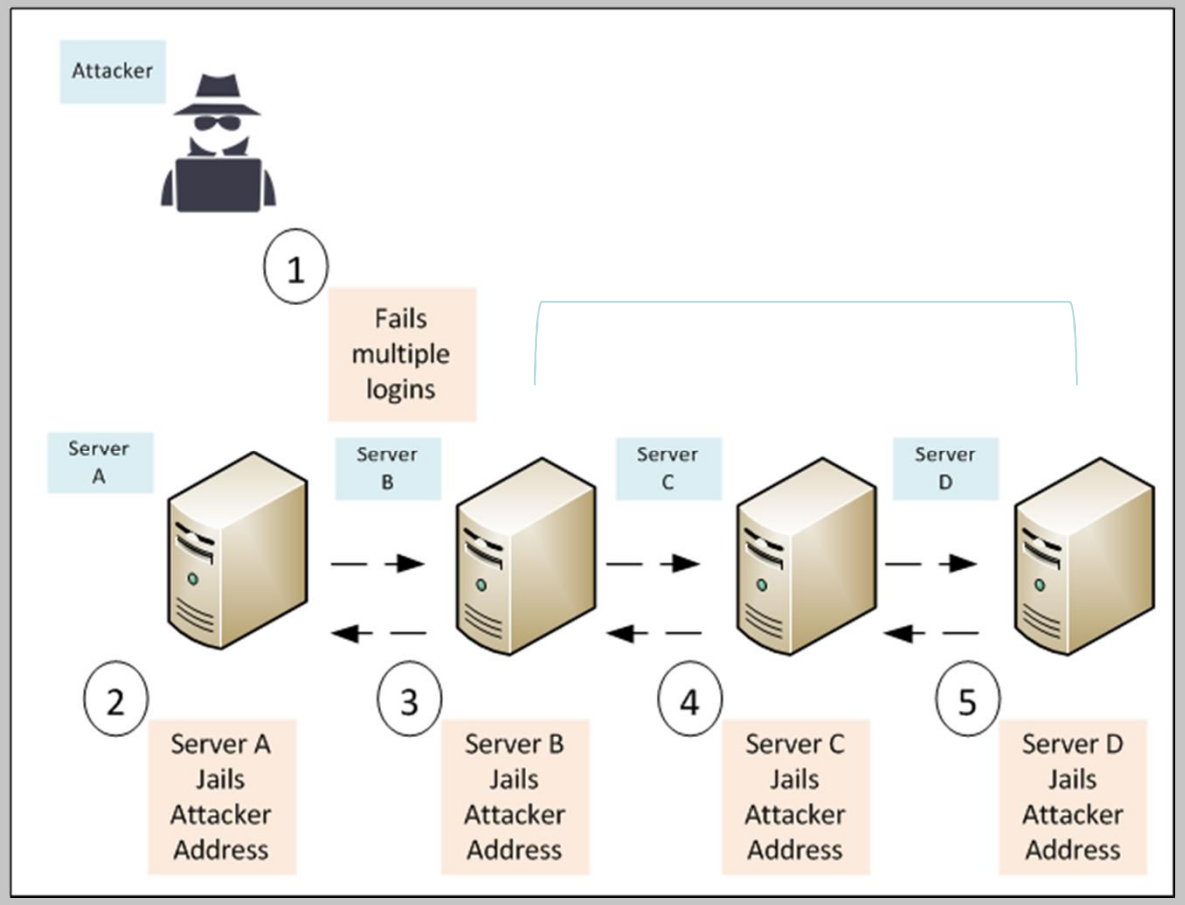
iptables

- Pulls Events
  - Reads Threat Events from ANP
  - NATs Threats from Local Webserver to Local Honeypot
- High Interaction Honeypot of Your Website?
  - Log Their Activity
  - Include a beacon?

Sharing Also Provides

- Increased Visibility
  - We don't change our enterprise
  - Everything Keeps Doing Its Job
  - We are giving them greater visibility to do so
- Ability to Be Proactive

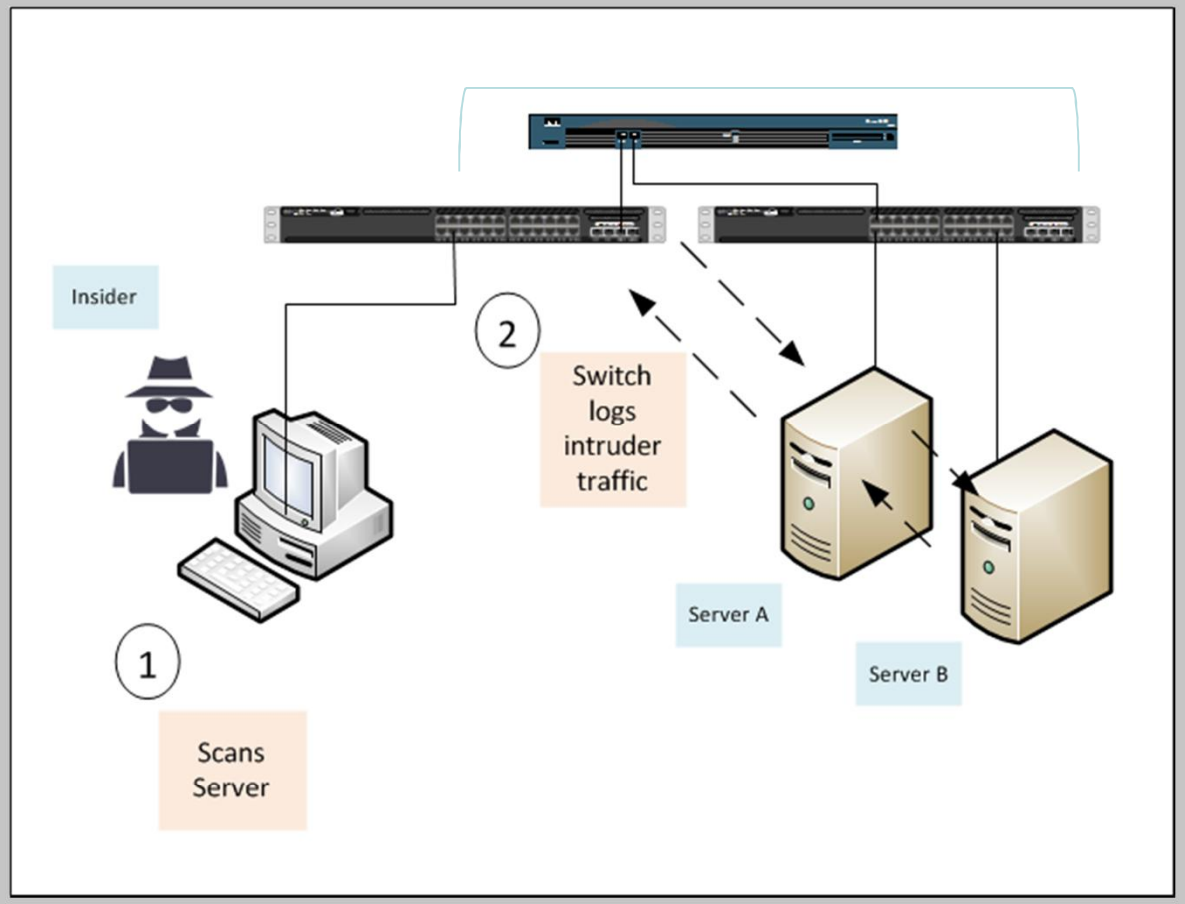
Expanded Visibility



Emerges With Sharing

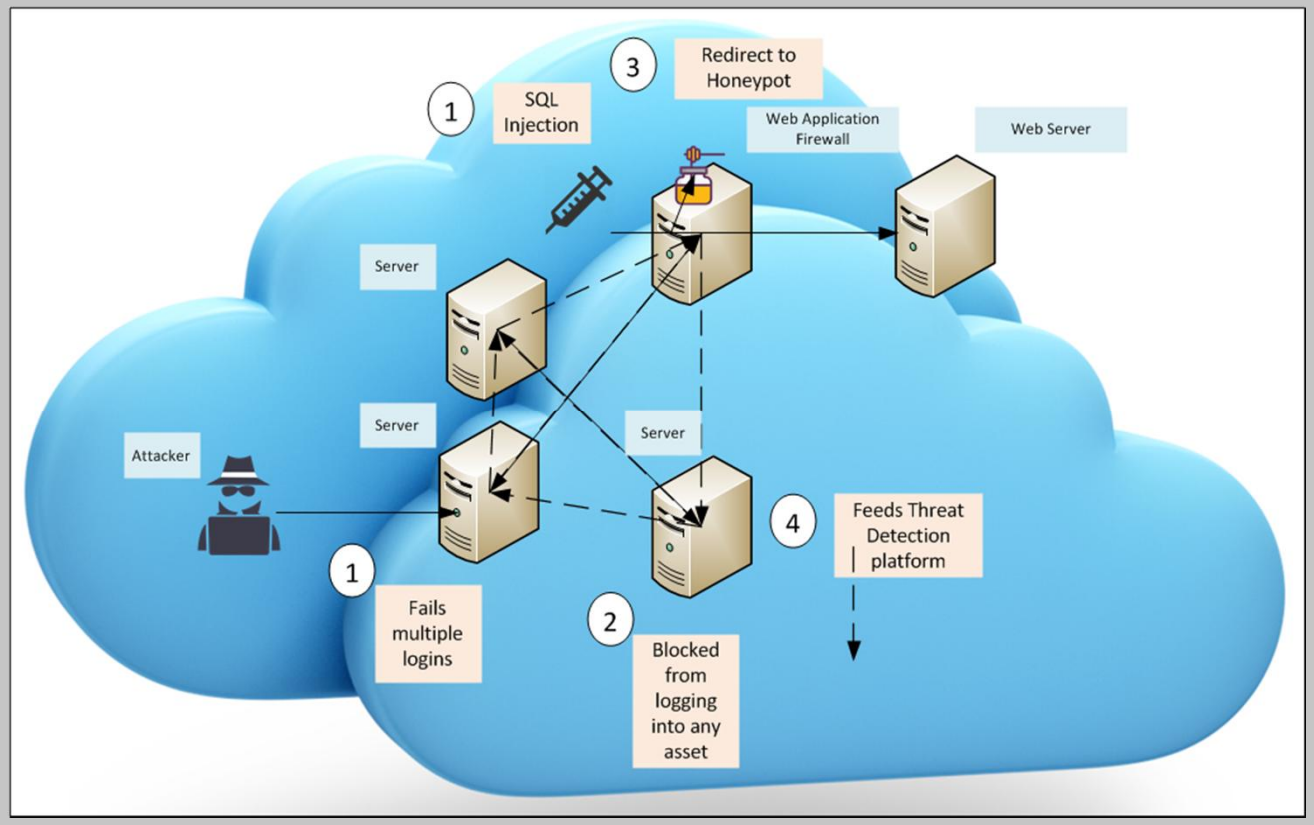
- Cooperative Behavior
- Ability for the Enterprise To Act On Its Own

Cooperative Behavior

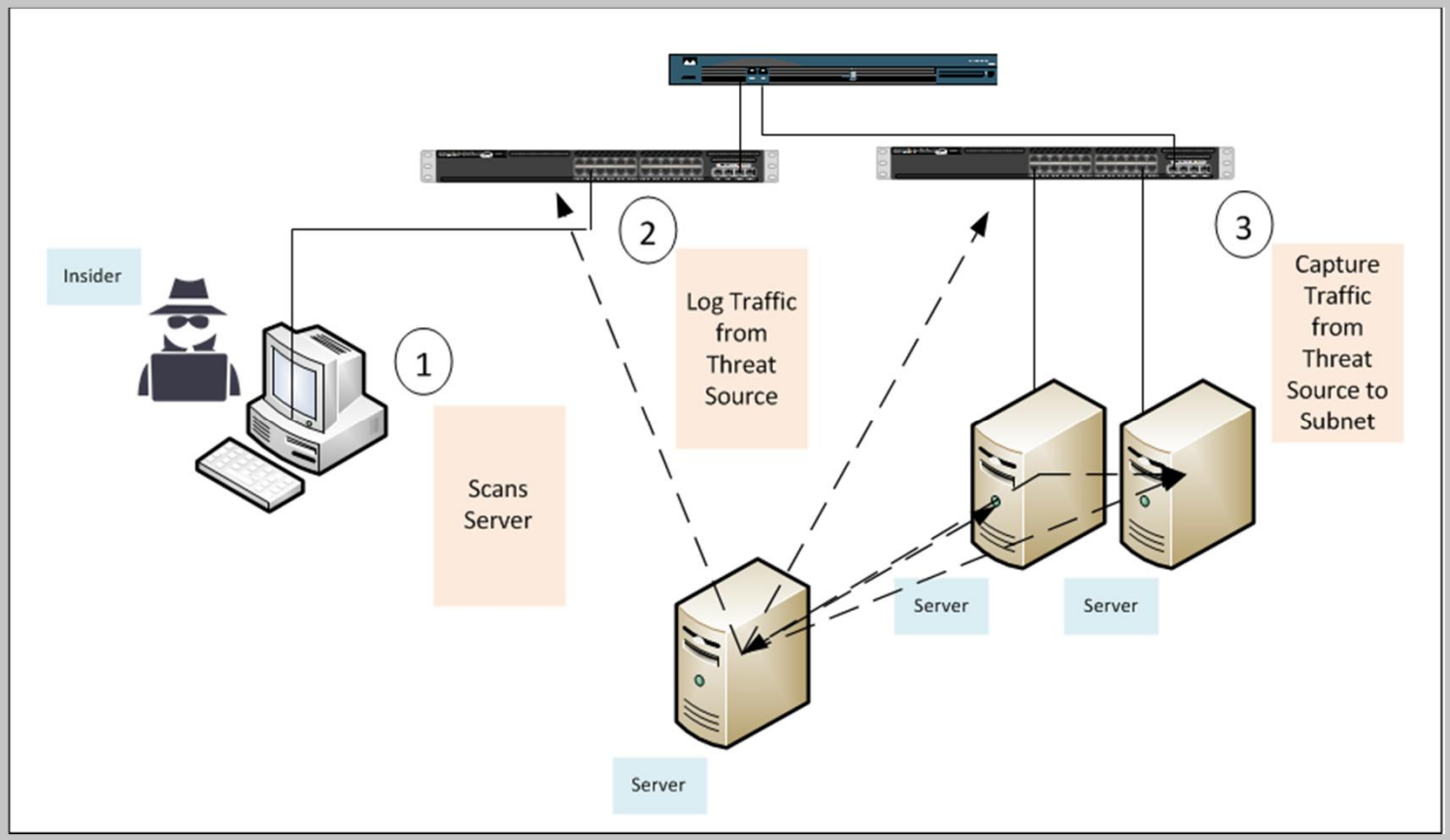


Building Skynet

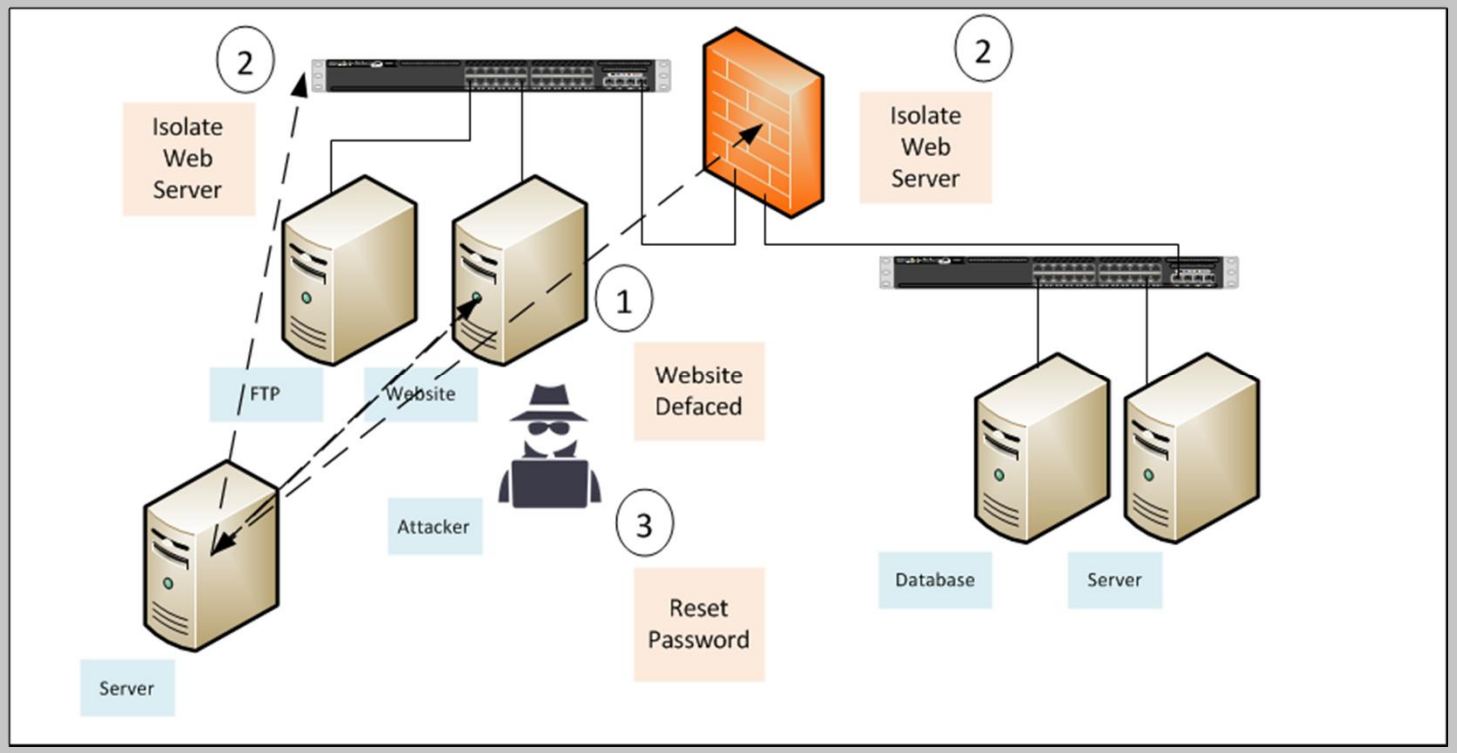
Acting to Defend The Network



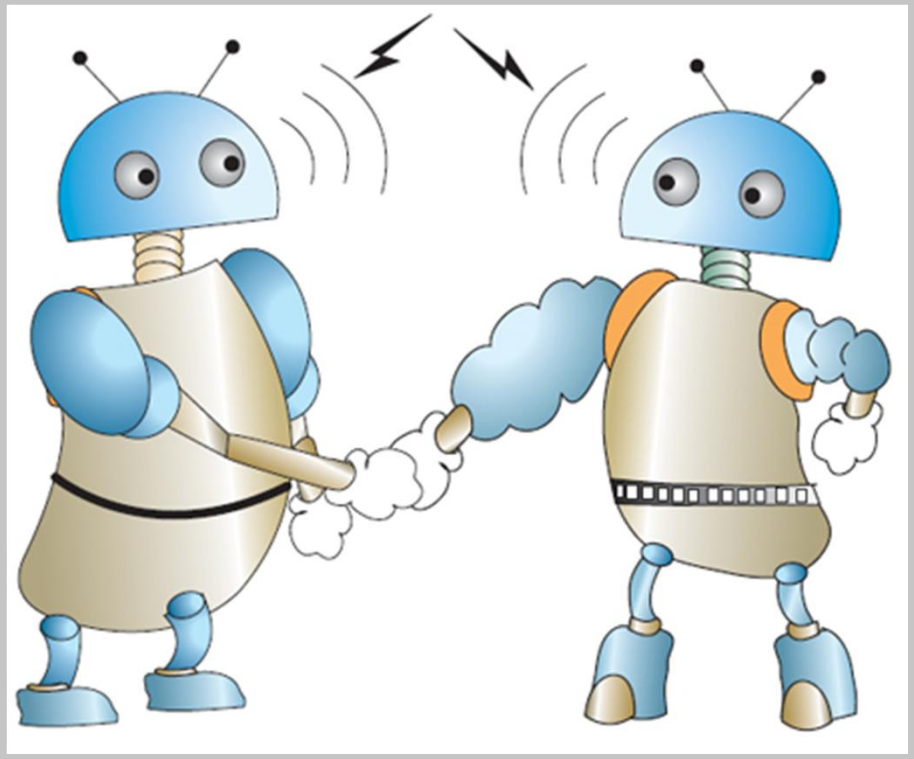
Acting To Investigate A Threat



Acting To Respond To An Incident



Demonstrations



Our Systems

Acting To Defend The Network

System	Cloud	Using
anp1	Amazon	anp,fail2ban4anp,modsec4anp
anp2	Azure	anp,fail2ban4anp,blacklist4anp
anp3	Azure	anp,fail2ban4anp,modsec4anp,iptables4anp

Acting to Defend The Network

Making It Better

Needed Improvements

- Additional Message Types
  - Add Target Event
  - Remove Target Event
- More Interfaces!
- Peer Groups
- Filters for Peers and Messages

Future Direction

- Internet of Things
- Reporting Events
- Export to STIX/TAXII

Making The Difference

- Machine To Machine Communication Solves Many Problems
- It Doesn't Have To Be The Apocalypse
- With It We Can
  - Get To The Threat On Time
  - Make Sure Evidence is Captured
  - Make Sure That The Threat Is Stopped
- We Can Do It With A Limited Staff

Final Thoughts

- Its Common To Kill Problems with Money and People
- Understanding Your Problem Means Better Results
- Enabling Synergies
  - Self Defending Networks
  - Self Investigating Networks
  - Self Responding Networks

Adaptive Network Protocol (ANP)

Adaptive Network Protocol (ANP) Agent v1.0.0

- Free and open-source solution for sharing events between systems
- Allows your systems to respond to threats in coordinated manner
- Share events both locally and remotely
- Python-Based

SHA1 hash is **976b9e004641f511c9f3eef770b5426478e8646a**
Updates can be found at https://adaptive-network-protocol.sourceforge.io/

Blacklist

Adaptive Network Protocol (ANP) Interface for Blacklists v1.0.0

- Free and open-source solution for generating a blacklist from shared events
- Pulls events from ANP writes them to blacklist
- Native Interface
- Python-Based

SHA1 hash is **6fdf91572909e97c5f6e005c93da0524a03463c8**
Updates can be found at https://adaptive-network-protocol.sourceforge.io/

Fail2Ban

Adaptive Network Protocol (ANP) Interface for Fail2Ban v1.0.0

- Free and open-source solution for jailing the source of threat events
- Pulls events from ANP and implements a jail in Fail2Ban
- Publishes events to ANP for sharing
- Native Interface
- Python-Based

SHA1 hash is **5c210858b5711d326bf1740620df4dedfe7a69c9**
Updates can be found at https://adaptive-network-protocol.sourceforge.io/

iptables

Adaptive Network Protocol (ANP) Interface for iptables v1.0.0

- Free and open-source solution for NATing the source of threat events toward your honeypot
- Pulls shared events from ANP and implements NAT in iptables
- Native Interface
- Python-Based

SHA1 hash is **5c210858b5711d326bf1740620df4dedfe7a69c9**
Updates can be found at https://adaptive-network-protocol.sourceforge.io/

modsec

Adaptive Network Protocol (ANP) Interface for modsec v1.0.0

- Free and open-source solution for sharing the source of threat events
- Publishes events to ANP for Sharing
- Native Interface
- Python-Based

SHA1 hash is **5c210858b5711d326bf1740620df4dedfe7a69c9**
Updates can be found at https://adaptive-network-protocol.sourceforge.io/

Links

- https://cybersponse.com/
- https://www.hexadite.com/
- https://www.phantom.us/
- https://www.siemplify.co/
- https://www.fireeye.com/products/security-orchestrator.html
- https://swimlane.com/
- https://www.saas-secure.com/online-services/fail2ban-ip-sharing.html
- http://www.blocklist.de/en/download.html
- https://www.blackhillsinfosec.com/configure-distributed-fail2ban/
- https://stijn.tintel.eu/blog/2017/01/08/want-to-share-your-fail2ban-ip-blacklists-between-all-your-machines-now-you-can
- https://serverfault.com/questions/625656/sharing-of-fail2ban-banned-ips
- https://github.com/fail2ban/fail2ban/issues/874


Links

- https://superuser.com/questions/940600/iptables-redirect-blocked-ips-from-one-chain-to-a-honeypot
- http://cipherdyne.org/psad/
- https://taxiiproject.github.io/
- https://stixproject.github.io/


Q&A