splunk> .conf2017

Tossing Splunk in Your PAN

Ninja's Guide to the Galaxy of Splunk and Palo Alto Networks

Kevin Gonzalez | Security Operations Manager

September 25, 2017 | Washington, DC

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

LENNAR®

Agenda

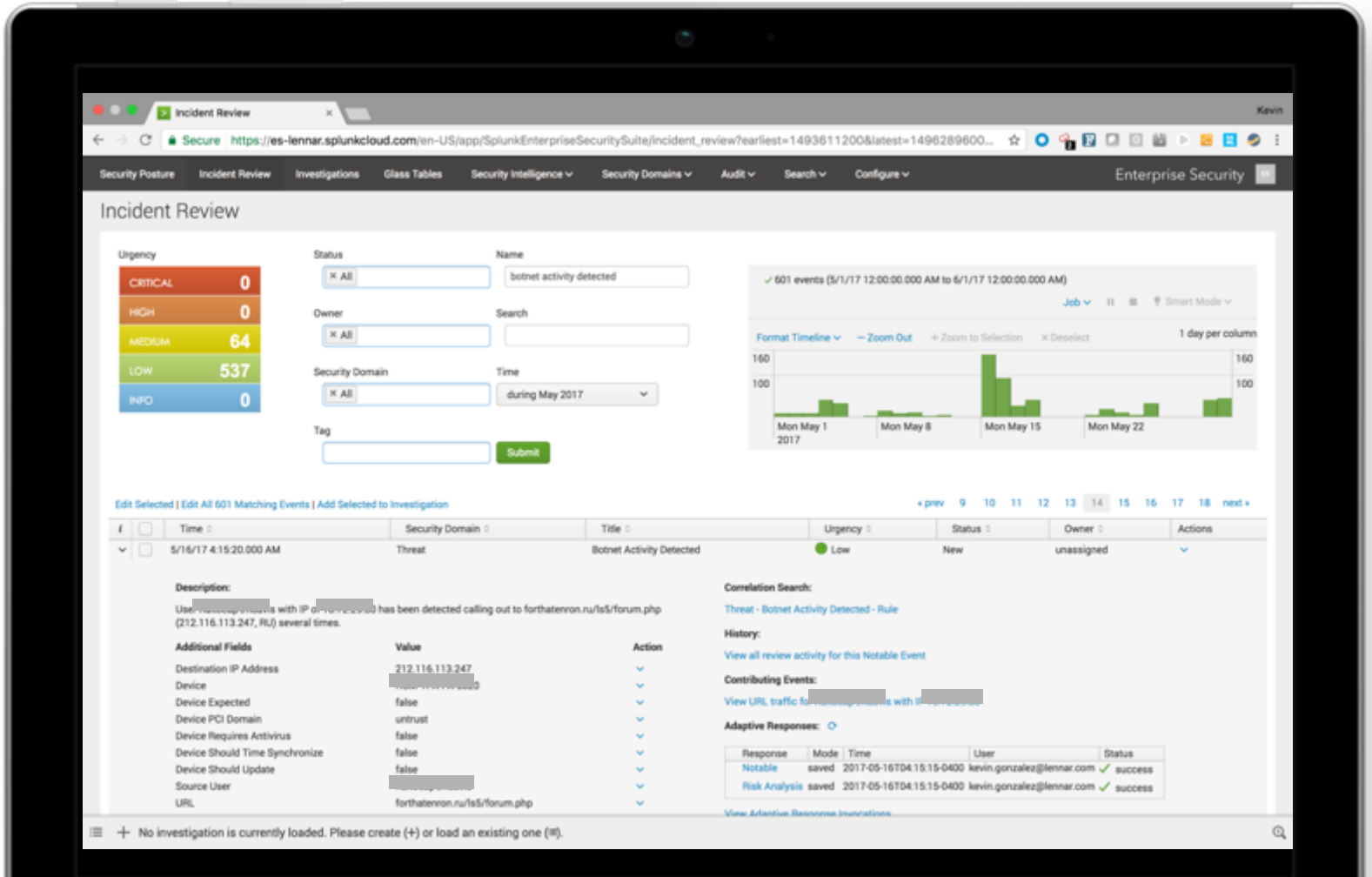
1. Utilizing Splunk Enterprise Security to:
   - Reduce alert chaos
   - Tame your PANW Threat Intelligence Feeds
2. Saving time with a Splunk/PANW API Fusion
3. Knowing the “who” at all times by populating PANW’s User-ID
4. Utilizing the Splunk Universal Forwarder to fix all of your problems

Splunk Enterprise Security & Palo Alto Networks

- Incident Response
  - Single pane of glass
  - Adaptive response based on correlated information
- Threat Intelligence
  - Centralized repository
  - Easy to maintain
  - Deduplication and content filtering

Splunk Enterprise Security & Palo Alto Networks

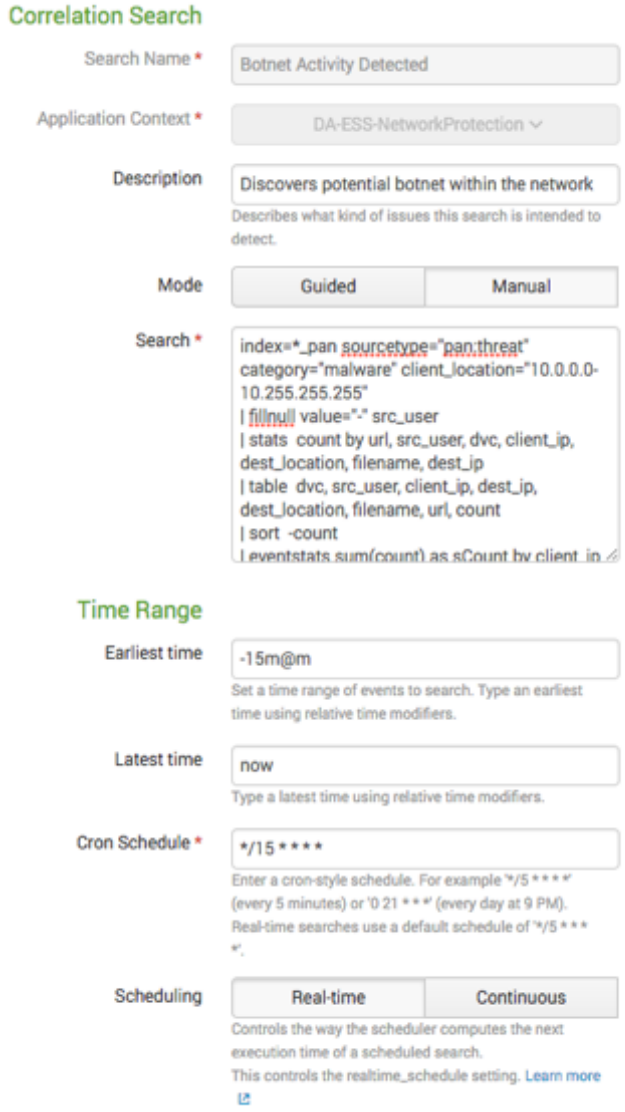
Integrating PANW into Enterprise Security

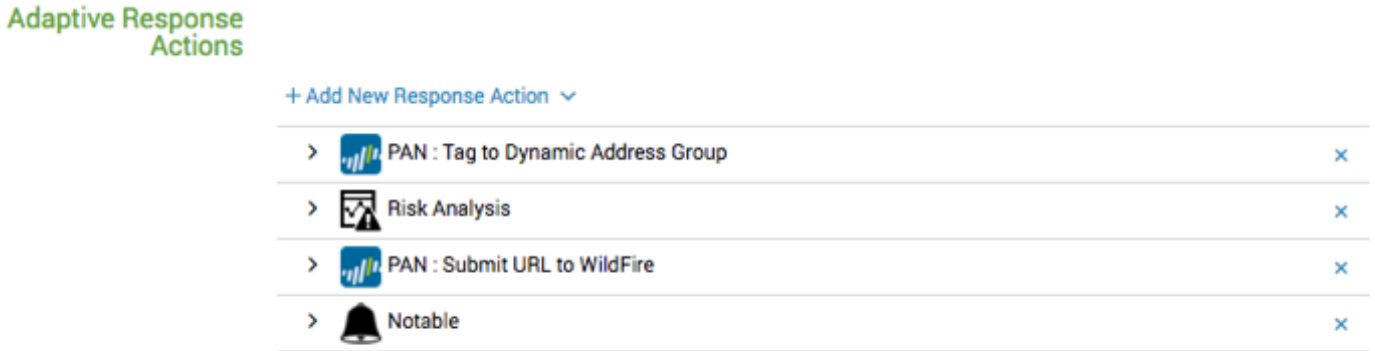


- Create Notable Events geared towards your PAN Firewall
  - Auto-Wildfire submissions
  - Per-Notable event tagging for PAN Dynamic Block Lists
- Utilize Threat Intelligence Data Models to feed your Palo Alto Network Firewalls:
  - De-duplicate and publish filtered threat intel
  - Auto-add trusted intel to dynamic block lists via tags

splunk> .conf2017

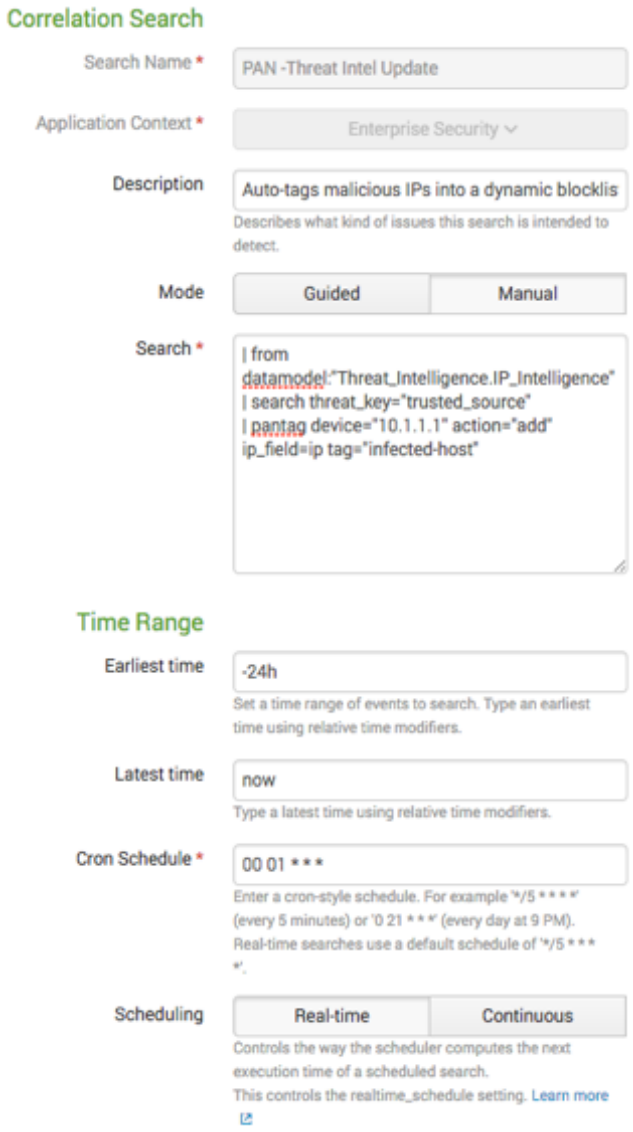
Splunk Enterprise Security & Palo Alto Networks






```
index=*_pan sourcetype="pan:threat" category="malware" client_location="10.0.0.0-10.255.255.255"
| fillnull value="-" src_user
| stats  count by url, src_user, dvc, client_ip, dest_location, filename, dest_ip
| table  dvc, src_user, client_ip, dest_ip, dest_location, filename, url, count
| sort  -count
| eventstats sum(count) as sCount by client_ip
| where sCount > 4
| lookup ut_parse_extended_lookup url
| fields - sCount
```

Splunk Enterprise Security & Palo Alto Networks




```
| from datamodel:"Threat_Intelligence.IP_Intelligence"
| search threat_key="trusted_threat_source"
| pantag device="x.x.x.x" action="add" ip_field=ip tag="no_bueno"
```

pantag

http://pansplunk.readthedocs.io/en/latest/commands.html#pantag

pantag

The `pantag` command shares context with the firewall by tagging IP addresses found in Splunk into Dynamic Address Groups.

Command added in App version 4.1. New parameters added in App version 5.0.

Syntax:

```
pantag device=<hostname>|panorama=<hostname>
[serial=<serial-of-device-in-panorama>] [vsys=<vsys#>]
[action=<add|remove>] [ip_field=<field-containing-IPs>]
tag=<tag>|tag_field=<field-containing-tags>
```

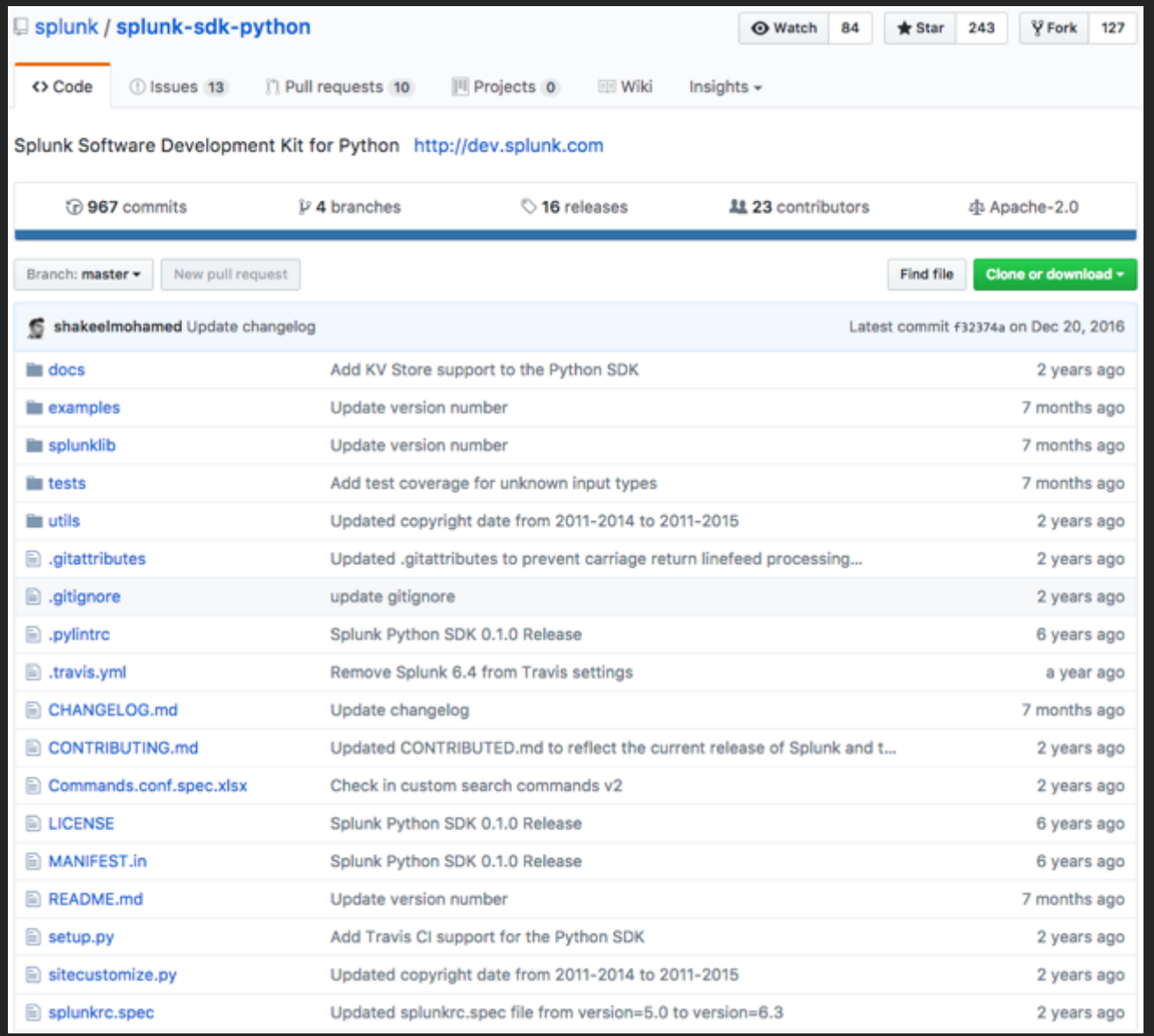
Parameter	Default	Added in	Usage
device		4.1	IP or hostname of firewall
panorama		5.0	IP or hostname of Panorama
serial		5.0	Serial of firewall (required if using panorama parameter
vsys	vsys1	5.0	VSYS ID (eg. vsys2)
action	add	4.1	Add or remove the tag
field	src_ip	4.1	Same as ip_field parameter (deprecated in 5.0, use ip_field)
ip_field	src_ip	5.0	Log field containing IP address to tag
tag		4.1	Tag for the IP, referenced in the Dynamic Address Group
tag_field		5.0	Log field containing the tag for IP address in the same log

API Harmony

Splunk, Palo Alto Networks, and their API's

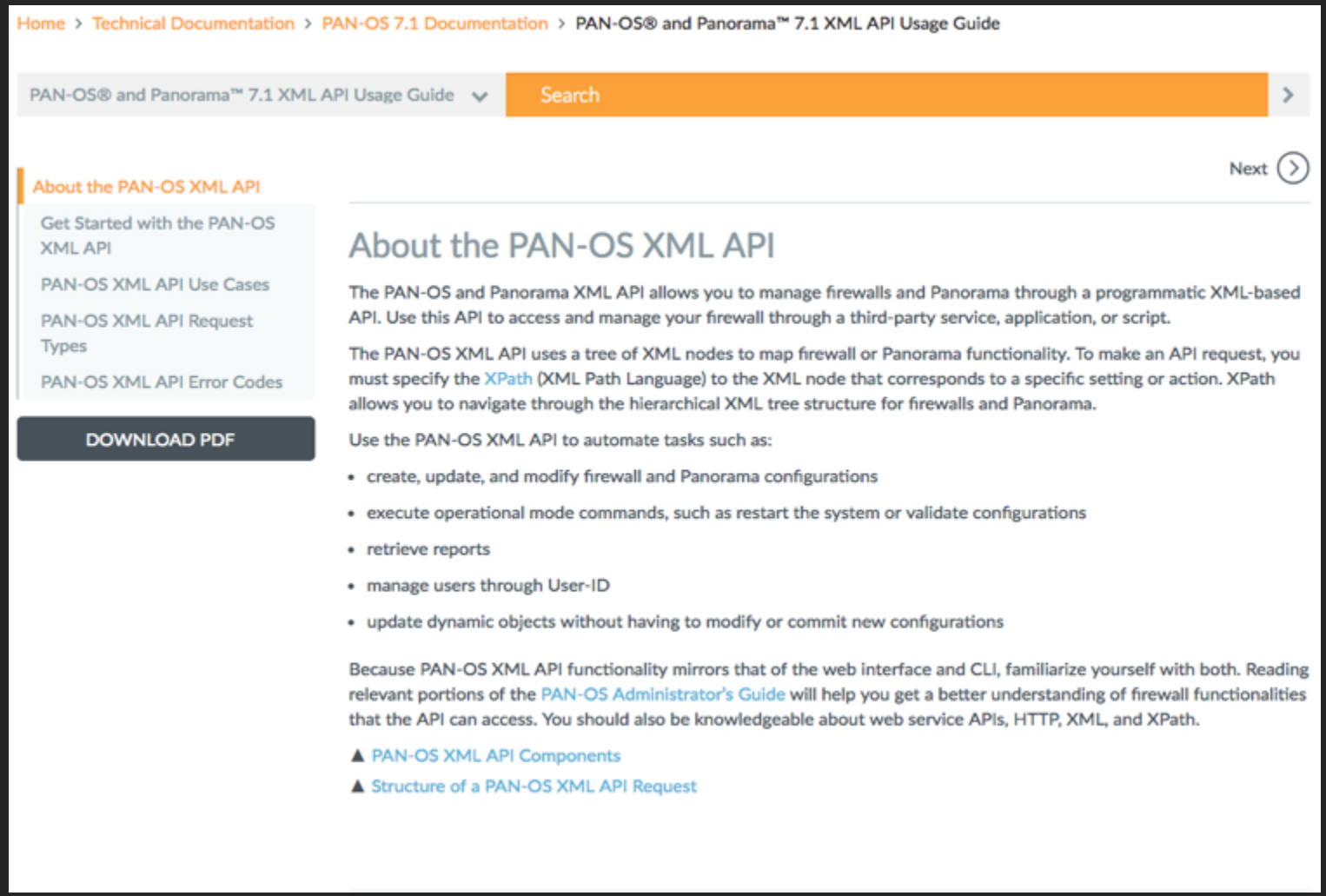
Splunk SDK for Python

https://github.com/splunk/splunk-sdk-python



PAN-OS XML API

https://www.paloaltonetworks.com/documentation/71/pan-os/xml-api



splunk> .conf2017

pantag - Extended

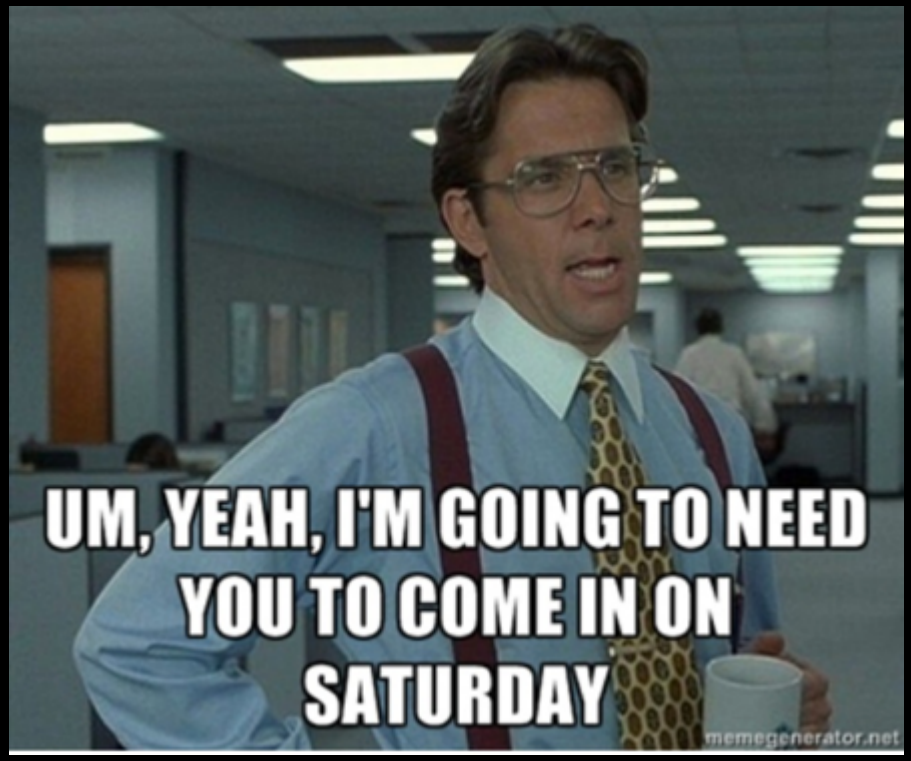
Auto-tag trusted Domain and HTTP IOCs to Dynamic Block Lists

- Use your Splunk Enterprise Security Data Models
- Splunk Cloud customers are currently limited to IP and HTTP

```
#!/usr/bin/env python
import splunklib.client as client
import requests
import re

service = client.connect(
    username='admin',
    password='password',
    host='10.1.1.2',
    port='8089')

urlstart = "https://10.1.1.1/?type=config&action=set&key=********&xpath=/config/shared/profiles/custom-url-category/entry[@name='Malicious-Domains']/list&element=<member>"
urlend = "</member>"

search = '| from datamodel:"Threat_Intelligence.HTTP_Intelligence" | search threat_key="trusted_threat_source" | table threat_key url'
results = service.jobs.oneshot(search)

rexRemover = re.compile("\s+|\<value\>\<text\>|\<\/text\>\<\/value\>")

for line in results:
    match = re.search("\<value\>\<text\>", line)
    if match:
        value = rexRemover.sub('', line)
        urlMod = urlstart + value + urlend
        requests.get(url=urlMod, verify=False)

requests.get(url="https://10.1.1.1/api/?type=commit&&key=********&cmd=<commit></commit>" , verify=False)

```

How About Using the APIs to Maintain Your Address Objects?

- Pre-existing address object library? No problem.
- Need it done now? No problem.
- Create new address object library managed by Splunk data
- Regularly update address objects with pre-defined tags



Updating PAN-OS Address Objects

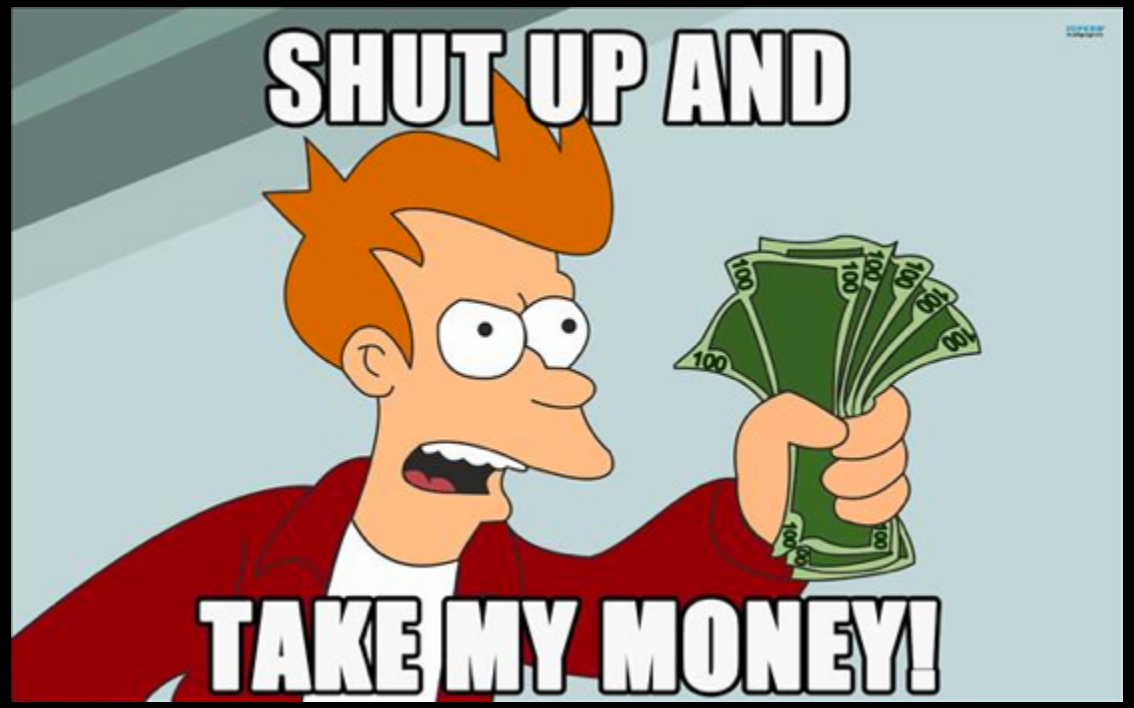
```python
#!/usr/bin/env python
import splunklib.client as client
import requests
import re

# function to rename address objects
def pan_name_update (adDict, apiKey):
    url = None
    tagStripper = re.compile("\=\<ip\-netmask\>(\d{1,3}\.){3}\d{1,3}\/32\<\/ip\-netmask\>")
    for dictObj in dictList:
        host = dictObj['host']
        ip = tagStripper.sub('', dictObj['ip'])
        url = "https://10.1.1.1/api/?type=config&action=rename&xpath=/config/shared/address/entry[@name='%s']&newname=%s&%s" % (ip,
        requests.get(url=url, verify=False)

# function to set address objects with pan
def pan_tag_update (adDict, apiKey):
    url = None
    host = adDict['host']
    ip = adDict['ip']
    tags = adDict['tags']
    spaceStripper = re.compile("^\s+")
    # validate whether address object already exists, if not create it
    url_prelim = "https://10.1.1.1/api/?type=config&action=show&xpath=/config/shared/address/entry[@name='%s']&%s" % (host, apiKey)
    response = requests.get(url=url_prelim, verify=False)
    response_data = response.content.split('\n')
    if re.search("No\ssuch\snode", response_data[0]):
        url = "https://10.1.1.1/api/?type=config&action=set&xpath=/config/shared/address/entry[@name='%s']&element=%s%s&%s" % (host
        requests.get(url=url, verify=False)
    else:
        # iterate through current tags
        current_tags = None
        desc = ""
        for resp_obj in response_data:
            match_ip = re.search("\<\ip\-", resp_obj)
            if match_ip:
                pan_ip = spaceStripper.sub('', resp_obj)
                print pan_ip
            match_desc = re.search("\<description\>", resp_obj)
            if match_desc:
                desc = spaceStripper.sub('', resp_obj)
                print desc
            match_mem = re.search("\<member\>", resp_obj)
            if match_mem:
                tag_member = spaceStripper.sub('', resp_obj)
                if current_tags == None:
                    current_tags = "%s" % tag_member
                else:
                    current_tags = "%s%s" % (current_tags, tag_member)
        current_tags = "<tag>%s</tag>" % current_tags
        print current_tags
        if ip != pan_ip or tags != current_tags:
            url = "https://10.1.1.1/api/?type=config&action=edit&xpath=/config/shared/address/entry[@name='%s']&element=<entry name
            requests.get(url=url, verify=False)

# connect to splunk
service = client.connect(
    username = 'admin',
    password = 'password',
    host = '10.1.1.2',
    port = '8089' )

# define static arguments for splunk and pan
apiKey = 'key=AFJHIAEIAHFIOIIOFOQIJOFASFJOI@KJOFIJAFKJJRJASIAJSFAJFLKFKLJIFOJQIFMKLAMFMIEOQPFMA'
kwargs_oneshot = {"count": 0}

# oneshot for ip to host address group mappings with tags
search = 'search index=*_scripts sourcetype=script:winsocreset "server" earliest=-24h | eval host=upper(host) | rex field=_raw "OS\
results = service.jobs.oneshot(search, **kwargs_oneshot)
```

- Address object:
  - Standardization
  - Creation
  - Modification
- Splunk query that contains all necessary data
- + crontab

splunk> .conf2017

Palo Alto Networks User-ID

Why we all want it…



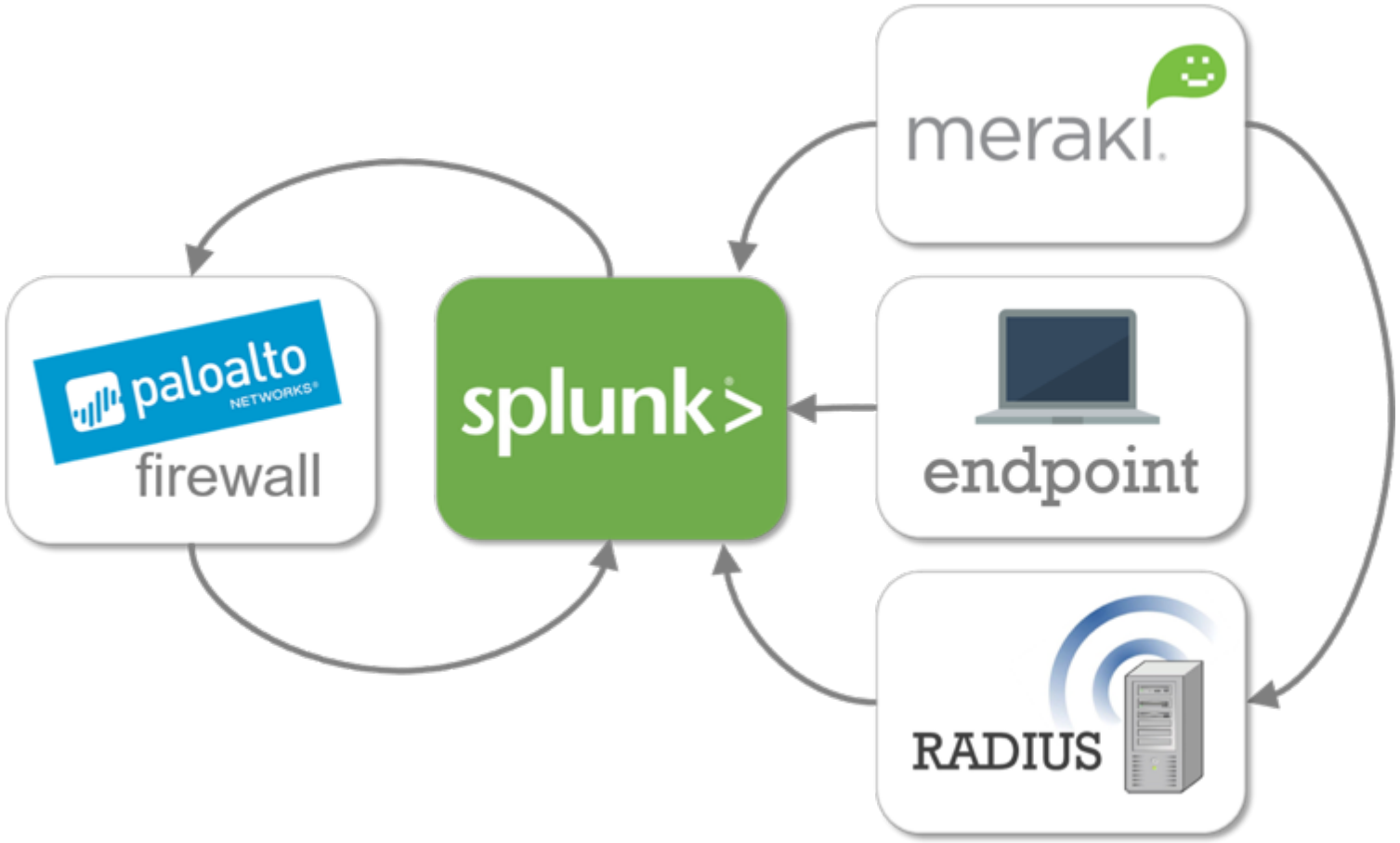
- Benefits
  - Improved visibility
  - Policy Control
  - Logging, Reporting and Forensics
- Use Cases
  - Security
  - Legal
  - Human Resources

Palo Alto Networks User-ID

Why we had some problems…

- Non-Microsoft endpoints
- No centralized point of reference for User-ID Agent
- Cloud-based systems
- Lack of historical data

Palo Alto Networks User-ID – Solution Overview



Updating User-ID With RADIUS Logs

```
index=meraki mac=* src=10.* src!=0.0.0.0
| eval src_mac=upper(replace(mac,":","-"))
| rex field=src "(?<src_ip>(\d{1,3}\.){1,3}\d{1,3})"
| dedup src_mac src_ip
| table src_mac src_ip
| outputlookup meraki_src.csv
```

```
index=wineventlog host=radius1 Client_IP_Address=* Security_ID!=*$ src_ip=*
| mvexpand Security_ID
| mvexpand src_ip
| dedup Security_ID src_ip
| search Security_ID!="NULL SID"
| rename Security_ID as user
| panuserupdate panorama=x.x.x.x serial=001234567890
| fields user src_ip status
```

- X amount of incomplete sources can depict a full picture
- Wireless AP Logs
  - Device MAC
  - Device IP
- RADIUS Logs
  - User
  - Device MAC
  - Automatic Lookups

panuserupdate

http://pansplunk.readthedocs.io/en/latest/commands.html#panuserupdate

panuserupdate

The `panuserupdate` command synchronizes user login events with Palo Alto Networks User-ID.
More information: User-ID with Splunk

Added in App version 5.0. For previous versions, refer to the panupdate command.

Syntax:

```
panuserupdate device=<hostname>|panorama=<hostname>
[serial=<serial-of-device-in-panorama>] [vsys=<vsys#>]
[action=<login|logout>] [ip_field=<field-containing-IPs>]
user_field=<field-containing-usernames>
```

Parameter	Default	Usage
device		IP or hostname of firewall
panorama		IP or hostname of Panorama
serial		Serial of firewall (required if using panorama parameter
vsys	vsys1	VSYS ID (eg. vsys2)
action	login	Tell the firewall user logged in or logged out
ip_field	src_ip	Log field containing IP address
user_field	user	Log field containing the username

Updating User-ID With SIP and “Best Guess” Scenarios

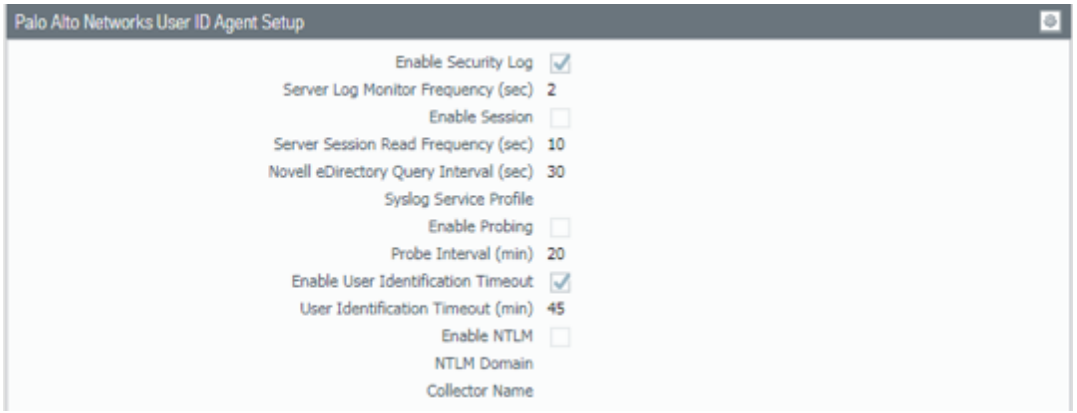
```
index=*_pan url="*/?sipuri=*"
| rex field=url "\/\?sipuri\=(?<user>[\w\-\.\@]+)"
| mvexpand user
| dedup user src_ip
| search user!=unknown
| panuserupdate panorama=x.x.x.x serial=001234567890
| fields user src_ip status
```

```
index=*_pan (user=unknown OR user="no user") ("@domain1.com" OR "@domain2.com")
| eval tmp=lower(_raw)
| rex field=tmp "\=(?<userTmp>[a-z\.]+@(domain1|domain2)\.com)"
| rex field=userTmp "(?<pre>[a-z\.]+)@(?<suf>(domain1|domain2)\.com)"
| eval user=pre+"@*"+suf
| dedup user src_ip
| panuserupdate panorama=x.x.x.x serial=001234567890
| fields user src_ip status
```

- Know your apps
- Know the data from those apps
- Now apply that knowledge
- Examples:
  - SIP Data from Lync/Skype
  - Raw URL Logs passing user info

“Best Guess” Scenarios and Timeouts…

```python
#!/usr/bin/env python
import splunklib.client as client
import requests
import re

# connect to splunk
service = client.connect(
    username = 'admin',
    password = 'password',
    host = '10.1.1.2',
    port = '8089' )

# define static arguments for splunk and pan
apiKey = 'key=ASFJKQNFOQNFOIASFIOEJOIFJOKVMKMFPINFPOIQNIPOFUNIPUWNFAKSKLNFPQWHFOINLKAFNP
kwargs_oneshot = {"count": 0}

# oneshot search
search = 'search index=*_pan (user=unknown OR user="no user") ("@domain1.com" OR "@domai
results = service.jobs.oneshot(search, **kwargs_oneshot)

# parse into list
resultList = ([])
for xmlObj in results:
    resultList.append(xmlObj)

# begin parse and pan user-id updates
count = 0
fullEntry = ''
rexStripper = re.compile("\s+|\<value\>\<text\>|\<\/text\>\<\/value\>")
for obj in resultList:
    match = re.search("\<field\sk\=\'user\'\>", obj)
    if match:
        user = rexStripper.sub('', resultList[count+1])
        ip = rexStripper.sub('', resultList[count+4])
        entry = '<entry name="%s" ip="%s" timeout="360"/>' % (user, ip)
        fullEntry = '%s%s' % (fullEntry, entry)
    count += 1

# append pre and post xml tags for user-id
xmlStart = '<uid-message><type>update</type><payload><login>'
xmlEnd = '</login></payload></uid-message>'
xmlString = '%s%s%s' % (xmlStart, fullEntry, xmlEnd)

# post user-id info
url = "https://10.1.1.1/api/?type=user-id&vsys=vsys1&cmd=%s&%s" % (xmlString, apiKey)
requests.get(url=url, verify=False)
```



- Individualized User-ID timeout
- Increased customization
- APIs are your best friend…

splunk> .conf2017

Updating User-ID with Custom Scripts

```
$os = (gwmi -Class win32_operatingsystem).caption
if($os -notmatch "server"){
    wmic computersystem get username
    $ipL = gwmi -Class win32_networkadapterconfiguration | ?{$_.defaultipgateway -ne $null} | select ipaddress
    foreach($a in $ipL){
        $ipA = @()
        $ipA += $a.ipaddress
        foreach($ip in $ipA){
            Write-Host "IP = $ip"
        }
    }
}
```

```
index=*_scripts sourcetype=Script:UserAffinity IP=*
| rex field=_raw "UserName\s+(?<user>\w+[\x5C]\w+)"
| rex max_match=5 field=_raw "IP\s\=\s(?<src_ip>(\d{1,3}\.){3}\d{1,3})"
| mvexpand src_ip
| dedup user src_ip
| fillnull value="No User" user
| panuserupdate panorama=x.x.x.x serial=001234567890
| fields user src_ip status
```

- Endpoints are your true source
  - Contain LAN/WAN IPs
  - Contain User Info
- The Splunk Universal Forwarder are also your best friend
- Historical evidence of all User/IP mappings… completed.

splunk> .conf2017

Other Splunk Universal Forwarder Use Cases

As it relates to Palo Alto Networks…

- GlobalProtect Install Validation
- Version Checks
- Setting Checks
- Assess your egress
- Bug Workarounds

GlobaLprotect Bug Workaround Utilizing The Splunk Universal Forwarder

- v3.0.x
- Ubiquitous VPN on Windows
- Ungraceful Network Disconnects
- Internal DNS Settings Frozen

```
$checkGP = gwmi -class win32_product | ?{$_.name -match "globalprotect"}

if($checkGP -ne $null){
    $n = $checkGP.name
    $v = $checkGP.version
    Write-Host "Name = $n"
    Write-Host "Version = $v"
    wmic computersystem get username

    if (!(Test-Connection -ComputerName validationurl1.com -Count 2 -ErrorAction SilentlyContinue) -and !(Test-Connection -ComputerName validationurl2.com -Count 2 -ErrorAction SilentlyContinue)) {
        Write-Host "External Test Completed. No Connection Found."
        if (!(Test-Connection -ComputerName validationIP1 -Count 1 -ErrorAction SilentlyContinue) -and !(Test-Connection -ComputerName validationIP2 -Count 1 -ErrorAction SilentlyContinue)) {
            Write-Host "Internal Test Completed. No Connection Found."
            C:\Winodws\System32\ipconfig.exe /all
            C:\Windows\System32\netsh.exe int ipv4 reset
            C:\Windows\System32\netsh.exe int ipv6 reset
            C:\Windows\System32\netsh.exe winsock reset catalog
            C:\Windows\System32\netsh.exe interface ip set dns name= "Wireless Network Connection" dhcp
            C:\Windows\System32\netsh.exe interface ip set dns name= "Local Area Connection" dhcp
        }
    }
}
```

Questions, Comments, Concerns?

Thank You

Don't forget to rate this session in the .conf2017 mobile app