Threat-Hunting

Queries & Log Patterns Pack for the F5 Breach

How to use

1. **Pick your Platform** - Splunk, Microsoft Sentinel, or Elastic.
2. **Copy the query** from the relevant card.
3. **Replace placeholders** like INDEX_F5, MGMT_CIDR, ALLOWED_ADMIN_IPS.
4. **Run as ad-hoc hunt first**; convert to scheduled alert after tuning.
5. **Correlate findings** with other infra (e.g., firewall, NetFlow, proxy).

1) Admin logins from new / unusual sources (time, geo, ASN)

Catches: First-time or out-of-pattern admin sign-ins to BIG-IP (GUI/SSH) → early presence testing.

Why it matters: Most intrusions show anomalous admin auth before config changes or data movement.

SPLUNK (SPL)

```
index=INDEX_F5 (sourcetype=f5:bigip:* OR sourcetype=syslog)
(("sshd" OR "httpd" OR apmd) AND ("Accepted" OR "authentication success" OR "login succeeded")) user=*
| eval hour=strftime(_time, "%H")
| iplocation src_ip
| stats min(_time) as first_seen max(_time) as last_seen values(host) as device by user src_ip Country
| eval is_new = if(first_seen >= relative_time(now(), "-24h@h"), 1, 0)
```

```
| where is_new=1 OR NOT match(Country, "ALLOWED_COUNTRIES_REGEX")
| table last_seen device user src_ip Country
| sort - last_seen
```

Microsoft Sentinel (KQL) – SSH/GUI via Syslog

```
let Recent = 1d;
Syslog
| where TimeGenerated > ago(Recent)
| where Facility in ("auth", "daemon") or ProcessName in ("sshd","httpd","apmd")
| where SyslogMessage has_any ("Accepted", "authentication success", "login succeeded")
| extend user = extract(@'user=([^\s]+)', 1, SyslogMessage)
| extend src_ip = extract(@'(?:rhost=|from )([0-9a-fA-F\.\:]+)', 1, SyslogMessage)
| project TimeGenerated, HostName, user, src_ip, SyslogMessage
| invoke _GetGeoInfo(src_ip)
| summarize by TimeGenerated, HostName, user, src_ip, Country, City
```

Elastic (KQL)

```
(event.dataset : "f5.bigip" OR process.name : ("sshd","httpd","apmd"))
AND message : ("Accepted" OR "authentication success" OR "login succeeded")
```

Tuning:

- Maintain a **baseline table** of admin IPs / countries seen in last 30d and alert on first-seen in last 24h.
- Add a filter to ignore bastion/VPN IPs in ALLOWED_ADMIN_IPS.

2) iControl REST (management API) from the internet / non-allowlisted sources

Catches: POST/PATCH/DELETE to /mgmt/ endpoints from outside your admin network.

Why it matters: Many post-ex actions leverage iControl REST for silent, scripted changes.

SPLUNK (SPL)

```
index=INDEX_F5 sourcetype=f5:bigip:http*
(uri_path="/mgmt/" OR uri_path="/mgmt/tm/*" OR uri_path="/mgmt/shared/*")
method IN ("POST","PATCH","DELETE")
NOT cidrmatch("MGMT_CIDR", src_ip)
| stats count by _time, host, src_ip, method, uri_path, http_user_agent, status
| where status IN (200,201,204)
| sort - _time
```

Microsoft Sentinel (KQL)

```
CommonSecurityLog
| where RequestURL startswith "/mgmt/"
| where HttpRequestMethod in ("POST","PATCH","DELETE")
| where not(ipv4_is_in_range(SourceIP, "MGMT_CIDR"))
| project TimeGenerated, DeviceName, SourceIP, HttpRequestMethod, RequestURL, RequestClientApplication, DestinationPort
```

Elastic (KQL)

```
url.path : (/mgmt/*) AND http.request.method : (POST OR PATCH OR DELETE)
AND NOT source.ip : MGMT_CIDR
```

Tuning:

- Add explicit allowlist of admin subnets and service accounts.

- Flag curl/python/go user-agents touching /mgmt/.

3) Config changes outside change window (LTM/APM/ASM/AFM)

Catches: create/modify/delete on critical objects when no change ticket is open.

Why it matters: Hands-on-keyboard changes often happen late at night or weekends.

SPLUNK (SPL)

```
index=INDEX_F5 sourcetype=f5:bigip:audit
("modify" OR "create" OR "delete")
AND ( "ltm " OR "apm " OR "asm " OR "afm " OR "sys " )
| eval hour=strftime(_time,"%H"), dow=strftime(_time,"%A")
| where (hour<7 OR hour>19) OR dow IN ("Saturday","Sunday")  /* adjust hours */
| table _time host user object cmd
```

Microsoft Sentinel (KQL)

```
Syslog
| where ProcessName == "tmsh" or SyslogMessage has_any ("modify","create","delete")
| where SyslogMessage has_any ("ltm ","apm ","asm ","afm ","sys ")
| extend hr = datetime_part("Hour", TimeGenerated), dow = format_datetime(TimeGenerated, 'dddd')
| where hr < 7 or hr > 19 or dow in ("Saturday","Sunday")
| project TimeGenerated, HostName, SyslogMessage
```

Elastic (KQL)

```
process.name : "tmsh" AND message : ("modify" OR "create" OR "delete")
AND message : ("ltm " OR "apm " OR "asm " OR "afm " OR "sys ")
```

Tuning:

- Join to your ITSM to suppress approved windows.

- Add a list of privileged objects (e.g., auth user, sys db, ltm rule).

4) iRules created/edited (possible traffic interception or token capture)

Catches: ltm rule create/modify.

Why it matters: Attackers can insert iRules to harvest headers/cookies or redirect flows.

SPLUNK (SPL)

```
index=INDEX_F5 sourcetype=f5:bigip:audit
("create ltm rule" OR "modify ltm rule")
| rex field=_raw "ltm rule\s+(?<rule_name>[^\s]+)"
| table _time host user rule_name _raw
```

Microsoft Sentinel (KQL)

```
Syslog
| where SyslogMessage has_any ("create ltm rule","modify ltm rule")
| extend rule_name = extract(@'ltm rule\s+([^\s]+)', 1, SyslogMessage)
| project TimeGenerated, HostName, rule_name, SyslogMessage
```

Elastic (KQL)

```
message : ("create ltm rule" OR "modify ltm rule")
```

Tuning:

- Diff the rule body from your golden repo (hash compare).
- Alert if rule names like tmp*, test*, or sudden HTTP::collect usage appear.

5) APM policy import/export or sudden edits

Catches: Policy imports/exports; large changes to access profiles.

Why it matters: APM is tied to SSO/identity—a prime target for stealing tokens.

SPLUNK (SPL)

```
index=INDEX_F5 sourcetype=f5:bigip:audit OR sourcetype=f5:bigip:apm
("apm policy import" OR "apm policy export" OR "modify apm policy")
| table _time host user _raw
```

Microsoft Sentinel (KQL)

```
Syslog
| where ProcessName in ("apmd","tmsh")
| where SyslogMessage has_any ("apm policy import","apm policy export","modify apm policy")
```

Elastic (KQL)

```
(event.dataset : "f5.bigip.apm" OR process.name : ("apmd","tmsh"))
AND message : ("apm policy import" OR "apm policy export" OR "modify apm policy")
```

Tuning:

- Alert when exports happen or when imports occur outside change window.
- Track who can import/export—restrict to a short list.

6) UCS / SCF archives created (data grab of full config)

Catches: tmsh save sys ucs <name> or save sys config file ...

Why it matters: A single UCS contains full device config + often secrets—perfect for exfil.

SPLUNK (SPL)

```
index=INDEX_F5 sourcetype=f5:bigip:audit
("save sys ucs" OR "save sys config file")
| rex field=_raw "save sys (ucs|config file)\s+(?<artifact>[^\s]+)"
| table _time host user artifact _raw
```

Microsoft Sentinel (KQL)

```
Syslog
| where SyslogMessage has_any ("save sys ucs","save sys config file")
| extend artifact = extract(@'(?:ucs|file)\s+([^\s]+)', 1, SyslogMessage)
```

Elastic (KQL)

```
message : ("save sys ucs" OR "save sys config file")
```

Tuning:

- Alert on new artifact names, off-hours creation, and rapid sequence (multiple UCS files).
- Combine with egress anomalies (next section).

7) Shell escape and suspicious tooling via tmsh run util bash

Catches: Attempts to run shell commands (curl/nc/wget/python) from BIG-IP.

Why it matters: Attackers use shell to pull payloads or exfil data.

SPLUNK (SPL)

```
index=INDEX_F5 sourcetype=f5:bigip:audit
("run util bash" OR "bash -c" OR "tmsh run util")
| table _time host user _raw
```

Microsoft Sentinel (KQL)

```
Syslog
| where SyslogMessage has_any ("run util bash","bash -c","tmsh run util")
| project TimeGenerated, HostName, SyslogMessage
```

Elastic (KQL)

```
message : ("run util bash" OR "bash -c" OR "tmsh run util")
```

Tuning:

- Add keyword hits for curl, wget, nc, python, openssl s_client, scp.
- Consider disabling bash access if your ops doesn't require it.

8) Egress spikes / new TLS destinations from the management VLAN

Catches: Large or unusual outbound flows from BIG-IP management.

Why it matters: Exfil or C2 from devices that normally don't talk out.

SPLUNK (NetFlow/Zeek)

```
index=INDEX_NET (sourcetype=zeek:conn OR sourcetype=netflow)
src_ip=MGMT_CIDR dest_ip!=10.0.0.0/8 dest_ip!=172.16.0.0/12 dest_ip!=192.168.0.0/16
| stats sum(bytes_out) as out_bytes, count as conns by src_ip, dest_ip, dest_port
| where out_bytes > 500000000 OR conns > 200   /* adjust thresholds */
| sort - out_bytes
```

Microsoft Sentinel (KQL / NSG / Zeek)

```
let thresholdBytes = 500000000;
AzureNetworkAnalytics_CL
| where SrcIp_s in (MGMT_CIDR_LIST)
| where not(DstIp_s in ("RFC1918_RANGES"))
| summarize out_bytes = sum(BytesSent_d) by SrcIp_s, DstIp_s, DstPort_d
| where out_bytes > thresholdBytes
```

Elastic (KQL)

```
source.ip : MGMT_CIDR AND NOT destination.ip : (10.0.0.0/8 OR 172.16.0.0/12 OR 192.168.0.0/16)
```

Tuning:

- Build a known-destinations list (support portals, update servers).
- Alert on new dest + volume + off-hours triad.

9) New local admin users / role changes

Catches: create auth user or modify auth user with role admin.

Why it matters: Quiet backdoor with persistent access.

SPLUNK (SPL)

```
index=INDEX_F5 sourcetype=f5:bigip:audit
("create auth user" OR "modify auth user")
| rex field=_raw "auth user\s+(?<user>[^\s]+)"
| search _raw="role admin"
| table _time host user _raw
```

Microsoft Sentinel (KQL)

```
Syslog
| where SyslogMessage has_any ("create auth user","modify auth user")
| where SyslogMessage has "role admin"
```

Elastic (KQL)

```
message : ("create auth user" OR "modify auth user") AND message : "role admin"
```

Tuning:

- Alert on any new admin unless a change ticket exists.
- Track password resets & SSH key additions.

10) Cloud token/API key use from new regions after F5 events

Catches: Service account or app sign-ins (often tied to APM/integrations) from new geos.

Why it matters: If device-stored secrets are lifted, cloud access often follows.

SPLUNK (OKTA – if applicable)

```
index=INDEX_OKTA sourcetype=okta:system
eventType=authentication.session.start
user.displayName="svc_f5" OR user.displayName="apm_*"
| iplocation client.ipAddress
| stats values(client.ipAddress) as ips values(Country) as countries by user.displayName
| ...  /* compare to 30d baseline via summary index */
```

Microsoft Sentinel (Entra IS Sign-in Logs)

```
let Lookback = 30d;
let Recent = 1d;
let serviceAccounts = dynamic(["svc_f5","apm_*","SERVICE_ACCOUNT_PATTERN"]);
let baseline =
SigninLogs
| where TimeGenerated between (ago(Lookback) .. ago(Recent))
| where UserDisplayName has_any (serviceAccounts)
| summarize seenCountries = make_set(LocationDetails.countryOrRegion) by UserId;
SigninLogs
| where TimeGenerated > ago(Recent)
| where UserDisplayName has_any (serviceAccounts)
| lookup baseline on UserId
| where LocationDetails.countryOrRegion !in (seenCountries)
| project TimeGenerated, UserDisplayName, AppDisplayName, IPAddress,
Country=LocationDetails.countryOrRegion
```

Tuning:

- Maintain a baseline of countries/ASNs per service principal.
- Alert on new geo within 24h of device admin events.

Disclaimer

The materials, documents, and tools shared by Point Break Security GmbH and The Daily Signal channel are provided free of charge for the purpose of supporting and educating our audience.

While every effort is made to ensure accuracy and usefulness, Point Break Security GmbH, The Daily Signal, and any individual involved in producing or distributing this content do not guarantee the quality, accuracy, or suitability of the information or materials provided.

We cannot be held responsible or liable for any damages, losses, or issues that may arise from the use, misuse, or interpretation of these materials.

Users are solely responsible for verifying that any scripts, configurations, or recommendations are compatible with their own environment and for thoroughly testing them in a controlled or non-production setting before applying them to live systems.

By using these materials, you acknowledge that you do so at your own risk and accept full responsibility for how they are implemented.