Continuous Security Monitoring in non-Active Directory Environments

Blair Gillam

Continuous Security Monitoring in non-Active Directory Environments

GIAC (GMON) Gold Certification

Author: Blair Gillam, blair@detnstudios.com
Advisor: Christopher Walker, CISSP, CISA
Accepted: February 9, 2019

Abstract

Active Directory-centric monitoring techniques, tools, and methodologies have dominated information security conferences in recent years. Many alternative centralized directory services, including FreeIPA and OpenLDAP, are found in modern enterprises. Diagnostic and performance monitoring for these alternatives is well documented; however, security-related events can be recorded in different formats and multiple locations across both directory servers and clients. This paper investigates continuous security monitoring techniques for FreeIPA that can be leveraged by defenders to analyze and visualize common directory service security events in non-Active Directory environments. It explores change detection rules that can be applied at the user, group, and directory levels and presents example security metrics for detecting anomalous activity.

1. Introduction

Incident responders continue to observe attackers targeting and using legitimate credentials for initial access and lateral movement within enterprise networks (FireEye, 2018). While attackers' use of legitimate credentials can blend in with normal day-to-day user activity, the attacker must still interact with an enterprise's centralized directory services. This activity provides defenders with an opportunity to detect malicious activity by leveraging Continuous Security Monitoring (CSM) techniques.

Modern enterprise defenders subscribe to the generally accepted principle that preventing security incidents is futile. Instead, they have adopted a detection-first mindset as evidenced by enterprises "moving away from prevention-only approaches to focus more on detection and response" (Gartner, 2017). This new focus has fueled an increase in publicly available CSM research focused on monitoring and detecting malicious activity.

A majority of publicly available CSM techniques, tools, and research involving centralized directory environments is focused on Active Directory (AD) due to the ubiquitous nature of AD in enterprise environments. BloodHound, a tool developed by @_wald0, @CptJesus, and @harmj0y, implements an offensive graph theory technique that culls AD for users, groups, and trusts of interest (Robbins, 2016). Pablo Delgado and Roberto Rodriguez (@Cyb3rWard0g) have both documented approaches using the Elastic Stack (Elasticsearch, Logstash, and Kibana) to monitor events from Active Directory and Windows systems for interesting user activity (Delgado, 2018; Rodriguez, 2018). The Active Directory Security website offers numerous resources on attacking as well as defending and detecting attacks against Active Directory (Metcalf, 2018).

There is a lack of techniques, tools, and resources for implementing similar detection strategies in non-Active Directory environments—specifically those that revolve around FreeIPA, OpenLDAP, and other Linux/Unix-based Lightweight Directory Access Protocol (LDAP) directory services. The non-Active Directory Continuous

Security Monitoring research presented in this paper is an initial attempt to achieve some parity with AD CSM techniques.

Historically, solvable but sometimes painful issues have appeared when monitoring non-AD directory services. These problems are classified into three distinct categories:

Archaic logging formats: OpenLDAP supports multiple overlays that implement logging differently. The *accesslog* overlay logs access events to the Directory Information Tree (DIT). Audit logging via the *auditlog* overlay in OpenLDAP saves directory changes in standard LDIF format and while this format makes it easy to undo changes, ingesting the change events into a SIEM can be difficult.

Disparate logging locations: In non-AD environments, directory events can be logged in multiple different locations across both the directory servers and the directory service clients. Depending on the specific non-AD implementation, directory-related events may be recorded across various services (e.g., directory server, Kerberos daemon, SSSD daemon) without any standard format.

Log volume: Centralized logging in Linux/Unix environments is typically implemented using a variant of syslog. Important directory events can be lost in a sea of messages.

1.1.1. Lab Setup

A virtual lab (Figure 1) was built using VMWare workstation 14 Pro that contained a single non-Active Directory test domain (ipa.gmonlab.local). The domain was created using FreeIPA—an open-source identity management solution for Linux that combines an LDAP-capable directory server, DNS server, Certificate Authority, and Kerberos daemon. FreeIPA was selected since its feature set more resembles Active Directory than a standard LDAP server. The domain consisted of primary (ipa01) and secondary (ipa02) FreeIPA directory servers, a domain-connected server (svr01), two workstations (wkst01 and wkst02), and a server (svr02) running the web application Moodle. Moodle was configured to perform authentication & authorization checks via LDAP. A centralized syslog/Elastic Stack server (log01) was set up to receive syslog and

directory server event logs from all systems in the domain. The operating system used on all servers and workstations within the lab was CentOS 7.

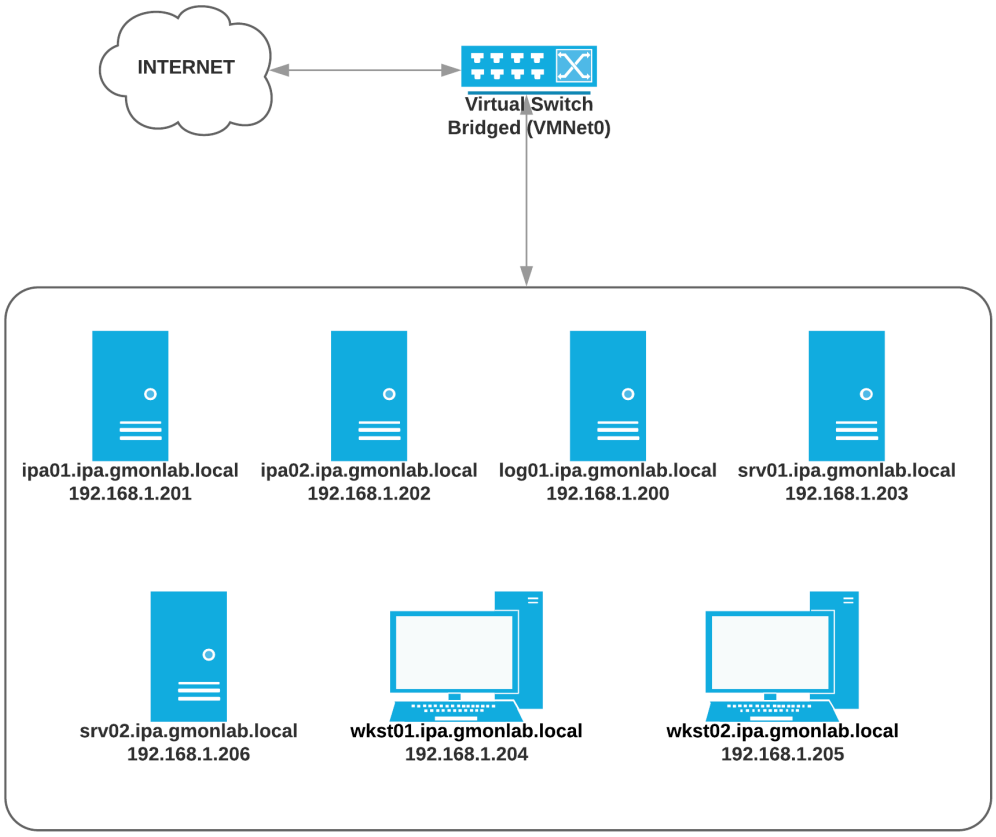

Figure 1: Lab Network Diagram

The primary purpose of the lab was used to log artifacts related to specific administrative tasks, user activity, and attacks expected to occur in non-Active Directory environments. The lab also served as a proving ground for determining security event logging requirements and developing methods for analyzing directory server events.

2. Directory Services Overview

2.1. Role of Directory Services

Centralized directory services contain "information on resources available on the network, including files, users, printers, data sources, and applications" as well as provide "users with access to resources and information" on the network (Newton, 2002). Common network services like Domain Name System (DNS), Network Information Service (NIS), and LDAP/X.500-based directory servers (Section 2.3) are all forms of directory services.

LDAP-specific directory services serve a critical role in modern enterprises by providing authentication (single sign-on), authorization (including role-based access control), and a centralized repository of information on company users, groups, and devices. Centralized management enabled by LDAP directory services decreases the administrative overhead when performing routine day-to-day tasks (e.g., performing employee onboarding and termination).

2.2. LDAP Standard

Lightweight Directory Access Protocol (LDAP) is a commonly used directory protocol that "defines a standard manner of organizing directory hierarchies and provides a standard interface for clients to access directory servers" (Newton 2002). LDAP was created by the University of Michigan to address the performance and complexity issues associated with the OSI Directory Access Protocol (DAP) (Fredriksson, 2010).

The LDAP protocol has been through several iterations since 1992 with the latest (LDAPv3) introduced as a standard in 2006. LDAPv3 fixes several critical issues including support for UTF-8 values (internationalization) and encrypted client/server communications using Simple Authentication and Security Layer (SASL).

2.2.1. LDAP Operations

LDAP is a request-response protocol where a client sends requests to the server which answers with a response. RFC 4511 documents the message formats for these operations, and RFC 4512 describes the information model (schema). According to RFC

4511 (Sermersheim, 2006), the LDAPv3 protocol supports ten different types of operations that are initiated by LDAP clients are:

BIND: Bind operations start an LDAP session and authenticate clients to the LDAP server. Both simple (e.g., anonymous, unauthenticated, or username/password) and SASL authentication methods are supported, and data confidentiality is provided by Transport Layer Security (TLS) or SASL (Harrison, 2006).

UNBIND: UNBIND operations terminate an LDAP session.

SEARCH: A client uses the SEARCH operation to request a set of entries matching the search filter from the directory. The server grants these requests if access controls allow the request to complete.

ADD: Clients can request to have entries added to the directory using ADD operations. The server ensures the request conforms to the directory's schema or other data models.

DELETE: DELETE operations remove entries from the directory. The server attempts to remove the entry as requested by the client but does not remove any of the entry's aliases.

MODIFY: MODIFY operations are used by clients to request alterations of directory entry contents. The server validates that the request conforms to schema and data model constraints before completing the modification.

MODIFY DN: Clients use the MODIFY DN operation when a Relative Distinguished Name (RDN) needs to be modified.

COMPARE: COMPARE operations determine if a specified directory entry has a given attribute value.

ABANDON: ABANDON operations are requested by a client that wants a server to stop processing a previously requested operation.

EXTENDED: The authors of the LDAPv3 protocol designed it to be extensible, and the EXTENDED operation is used to allow additional operations for services not yet

defined in the LDAP protocol. The STARTTLS operation that is used to initiate a TLS connection is an example of an EXTENDED operation.

2.3. Non-AD Directory Service Implementations

While LDAP is the standard protocol used among directory services, the directory service implementation details are the vendors' responsibility. This has led to implementations that all support the LDAP protocol but differ in features, functionality, and default configuration settings.

OpenLDAP is perhaps the most widely used open source implementation of LDAP and was based on The University of Michigan's Standalone LDAP Daemon and Standalone LDAP Update Replication Daemon (OpenLDAP Foundation, 2015). Variants of OpenLDAP have been created to serve different needs including ReOpenLDAP—a telco-oriented fork of OpenLDAP focused on scalability and redundancy (Yuriev, 2018)—and the LDAP Tool Box ("LTB") project which provides additional configuration and management functionality built-in to the default installation.

Commercial vendors have created directory server variants to serve enterprise customers. Oracle Internet Directory is an LDAPv3 compliant directory server that uses an Oracle database as the underlying datastore as an alternative to flat files or BerkleyDB files (Oracle, n.d.).

389 Directory Server (389ds)—formerly known as Fedora Directory Server—is a Linux-based, enterprise-class open source LDAPv3 server and is a development project within the RedHat ecosystem (Red Hat, n.d.). 389ds uses a slightly different underlying schema (strictly RFC 2252) than OpenLDAP which can lead to integration issues with 3rd party services that were tested with OpenLDAP.

Red Hat's Identity Management (IdM) and its upstream open source project (FreeIPA) are examples of directory services that bolt multiple components together to provide an easy to install and manage centralized directory solution (Atkisson, 2016). These "composite" servers combine services including an LDAPv3 compliant directory server (389ds), DNS, MIT Kerberos, role-based access control (RBAC), and tools for

provisioning and administration into a single solution that provides an Active Directory-like domain equivalent for Linux called an IPA domain.

2.4. Logging

The default logging configuration can significantly vary between different directory server implementations. For example, LDAP Tool Box (*openldap-ltb*) automatically configures syslog to receive logs on LDAP connections, operations, and results. OpenLDAP-base directory servers offer the ability to log access and directory changes using optional software components—known as overlays—that allow customization of the directory server's backend without requiring a custom backend. As previously mentioned, the *accesslog* overlay may be enabled that write out log events to the DIT, and the *auditlog* overlay can be used to log directory changes to LDIF-formatted files.

Logging in FreeIPA occurs on both the servers and clients. The server consists of multiple services with separate logging facilities. As a result, different log formats are used for the directory server (389ds), Kerberos daemon, Certificate Authority daemon, and Apache web server hosting the web user interface ("Web UI"). Clients connecting to an IPA domain may utilize the System Security Services Daemon (SSSD) for identity and authentication. When logging is enabled, SSSD logs events locally to a file on each domain member however authentication and authorization-specific log events are recorded in a different location.

3. Common Directory Service Events

Usage and change detection are core tenants for monitoring directory services. Whenever a directory entry (e.g., organization, user, or group) is added, deleted, modified, or used, events should be logged to create an audit trail of activity. Existing CSM research has created detailed lists of Windows event IDs that should be monitored to detect signs of compromise in Active Directory (Delgado, 2018; Microsoft, 2017). These events can be adapted into more generic CSM event categories that are specific for directory services.

Administrative Events: Administrative directory events encompass the creation, modification, and deletion of user accounts and groups. Group modifications are especially crucial as attackers can add users to administrative or otherwise privileged groups (e.g., role/host-based groups that allow administrative rights, SSH access, or sudo privileges) to gain additional access. CSM uses these events to detect persistence techniques (e.g., new malicious user accounts).

User Events: Individual user activity events (e.g., login activity, password changes, account lockout due to too many failed login attempts). User events are a crucial dataset for CSM that provide awareness into account activity. CSM uses these events to detect abnormal account activity and lateral movement between multiple endpoints.

Privileged User Events: Privileged accounts are user accounts that have been granted administrative rights to manage the directory. User impersonation, modifying log settings, and adding workstations to the domain are examples of privileged user events that should be monitored.

Service Account Events: Service accounts may be considered a type of privileged account that grants an application or service access to the directory. These accounts typically have read-only access to the directory and are used by applications to perform LDAP group membership lookups. From a security perspective, defenders should be suspicious of service accounts that are behaving abnormally by operating from unexpected endpoints or performing more intensive LDAP queries.

Directory Service Configuration Changes: Significant configuration changes—including disabling all logging, adding overlays, changing TLS certificates, and disabling insecure or secure transport mechanisms—to the directory service itself should rarely occur after a directory is deployed. All configuration change events should be examined to ensure security mechanisms are not being disabled. In practice, some of these changes can be difficult to instrument accurately due to cryptic log events.

Directory Server Trusts: Directory server trusts allow different domains to enable access to the domains' resources by authorized users. Trusts can be one-way or bi-directional (technically two one-way trusts) and the overall security of the domains are limited by the most poorly configured domain in the trust relationship. These events can be used to detect malicious attackers that leverage domain trusts to forge Kerberos tickets granting access to additional domain resources (Metcalf, 2015).

Blair Gillam;blair@detnstudios.com

4. FreeIPA Directory Event Logging

It is essential to understand what log artifacts are generated during various actions and attacks against a FreeIPA directory environment. While composite directory services make this more difficult due to multiple log locations and formats, artifacts were identified that defenders and event correlation systems could key upon for alerting.

4.1. Administrative Activity

Administrative activity is defined as changes to the users, groups, and configuration of the centralized directory service. This includes creating, updating, and deleting user accounts and groups, adding systems to the domain, and creating trust relationships between the IPA servers and an Active Directory domain. Events logged as a result of administrative actions are located within multiple log files on the FreeIPA directory servers:

Log Name	**File Location**
389ds Access Log	/var/log/dirsrv/slapd-<REALM>/access
389ds Errors Log	/var/log/dirsrv/slapd-<REALM>/errors
Kerberos Administration Server	/var/log/kadmind.log
Kerberos Key Distribution Center	/var/log/krb5kdc.log
FreeIPA Web UI Access Log	/var/log/httpd/access
FreeIPA Web UI Errors Log	/var/log/httpd/errors

Due to the request-response nature of LDAP, the 389ds access log records LDAP operations across multiple lines. The connection identifier ("conn") located in each line must be used to combine the LDAP operations into a single event. This can be used to track all LDAP operations that occur during each connection as well as identify who modified the directory entries.

Administrative tasks in FreeIPA are performed using the web user interface (Web UI) or the *ipa* command line tool (CLI). The python-based CLI tool uses the same API as the Web UI. As a result, administrative events are found in the 389ds logs and Web UI logs. The CLI tool was the primary method for interacting with the IPA directory servers. Before using the CLI tool, a directory administrator must authenticate to the Kerberos daemon using *kinit* before issuing the appropriate *ipa* command:

```
[root@ipa01 ~]# kinit admin
Password for admin@IPA.GMONLAB.LOCAL:
[root@ipa01 ~]# ipa user-add --first=Test --last=User5000 --cn="Test Use
r5000" --displayname="Test User5000" --gecos=tuser5000 --shell=/bin/bash
 --email="tuser5000@ipa.gmonlab.local" --random
User login [tuser5000]:
----------------------
Added user "tuser5000"
----------------------
  User login: tuser5000
  First name: Test
  Last name: User5000
  Full name: Test User5000
  Display name: Test User5000
  Initials: TU
  Home directory: /home/tuser5000
```

4.1.1. User Creation

When a user account is created, the 389ds access log and the Web UI access and error logs record these events. The 389ds access log contains a single LDAP ADD operation entry that discloses the new user account's distinguished name:

```
[12/Jan/2019:02:32:35.976912939 +0000] conn=341 op=5 ADD
dn="uid=scarr,cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local"
```

The Web UI access log does not contain any valuable data for a defender, but the Web UI error log provides a verbose single line entry:

```
[Sat Jan 12 02:32:35.982041 2019] [:error] [pid 9593] ipa: INFO:
[jsonserver_session] admin@IPA.GMONLAB.LOCAL: user_add/1(None,
givenname=u'Sigmund', sn=u'Carr', cn=u'Sigmund Carr',
displayname=u'Sigmund Carr', gecos=u'scarr', loginshell=u'/bin/bash',
mail=(u'scarr@ipa.gmonlab.local',), random=True, version=u'2.229'):
SUCCESS
```

The Web UI error log provides a more useful audit trail of user creation in a single log event. The standardized format includes who modified the directory (admin@IPA.GMONLAB.LOCAL), what modification occurred ("user_add"), and even

provides details about the new user. The Web UI error log provides value quickly with minimal analytical effort. The downside to the Web UI error log is that it does not provide enough information to determine where the user has connected to the directory service from. The Web UI and 389ds access logs provide the user's source IP address, but some additional analysis must be performed to map it back to the user creation event.

4.1.2. Group Creation

Group creation events are also logged in the 389ds access log and the Web UI error log. An example of the LDAP ADD operations observed in the 389ds access log is:

```
[12/Jan/2019:02:11:05.264391367 +0000] conn=659 op=2 ADD
dn="cn=testgroup2,cn=groups,cn=accounts,dc=ipa,dc=gmonlab,dc=local"
```

The Web UI error log provides additional details about the group creation event ("group_add"):

```
[Sat Jan 12 02:31:24.703148 2019] [:error] [pid 9535] ipa: INFO:
[jsonserver_session] admin@IPA.GMONLAB.LOCAL:
group_add/1(u'testgroup4', description=u'Test Group 4',
version=u'2.229'): SUCCESS
```

4.1.3. User Deletion

User deletion events use the LDAP DELETE ("DEL") operation. The following is an example entry from the 389ds access log:

```
[12/Jan/2019:14:02:03.769549762 +0000] conn=790 op=9 DEL
dn="uid=tuser1,cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local"
```

The Web UI error log entry records details on the recently deleted user account:

```
[Sat Jan 12 14:02:03.776178 2019] [:error] [pid 7720] ipa: INFO:
[jsonserver_session] admin@IPA.GMONLAB.LOCAL: user_del/1([u'tuser1'],
version=u'2.229'): SUCCESS
```

4.1.4. Group Deletion

Deleting a group uses the LDAP DELETE operation and results in a DEL entry within the 389ds access log, as shown below:

```
[12/Jan/2019:13:57:51.638042312 +0000] conn=782 op=8 DEL
dn="cn=testgroup2,cn=IPA.GMONLAB.LOCAL,cn=kerberos,dc=ipa,dc=gmonlab,dc
=local"
```

Once again, the Web UI error log provides a more verbose log entry showing the "group_del" function was used:

```
[Sat Jan 12 02:33:55.888209 2019] [:error] [pid 9593] ipa: INFO:
[jsonserver_session] admin@IPA.GMONLAB.LOCAL:
group_del/1([u'testgroup2'], version=u'2.229'): SUCCESS
```

4.1.5. Group Membership Modifications

Modifying a group's membership results in multiple log entries with the directory service. A SEARCH operation is performed for the user to be added or removed and is followed by an LDAP MODIFICATION ("MOD") operation. In the following example, the user "tuser3" was added to the group "testgroup1":

```
[12/Jan/2019:02:16:15.302517379 +0000] conn=663 op=2 SRCH
base="uid=tuser3,cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local"
scope=0 filter="(objectClass=*)" attrs=""
[12/Jan/2019:02:16:15.303120679 +0000] conn=663 op=3 MOD
dn="cn=testgroup1,cn=groups,cn=accounts,dc=ipa,dc=gmonlab,dc=local"
```

The Web UI's HTTPD error log contained a single log entry that included the group being modified, the user, and the action ("group_add_member"):

```
[Sat Jan 12 02:16:15.309948 2019] [:error] [pid 9535] ipa: INFO:
[jsonserver_session] admin@IPA.GMONLAB.LOCAL:
group_add_member/1(u'testgroup1', version=u'2.229', user=(u'tuser3',)):
SUCCESS
```

When a user is removed from a group, a MOD operation is logged in the 389ds access log. Unfortunately, only the group that was modified is logged:

```
[13/Jan/2019:21:06:01.057853630 +0000] conn=994 op=1 MOD
dn="cn=testgroup4,cn=groups,cn=accounts,dc=ipa,dc=gmonlab,dc=local"
```

However, the Web UI error log recorded a "group_remove_member" event with both the group modified and the user account that was removed:

```
[Sun Jan 13 21:06:01.058416 2019] [:error] [pid 25233] ipa: INFO:
[jsonserver_session] admin@IPA.GMONLAB.LOCAL:
group_remove_member(u'testgroup4', all=True, version=u'2.229',
user=(u'tuser5',)): SUCCESS
```

4.1.6. Workstation/Server Domain Membership

The previously described administrative activity—creating, deleting, and modifying user accounts and groups—are all examples of privileged user activity whereby specific accounts have additional permissions to manage an organization's assets and systems. Another privileged activity is the ability to add other systems ("domain members") to the IPA domain. In FreeIPA, domain members are typically added by installing the *freeipa-client* package and running the *ipa-client-install* CLI tool (Red Hat, 2018).

When a new host is joined to the domain using the *ipa-client-install* CLI tool, many entries are written to the 389ds access log, but a single LDAP ADD operation is recorded which includes the fully qualified domain (FQDN) of the new domain member:

```
 [14/Jan/2019:00:11:41.376580348 +0000] conn=1080 op=7 ADD
dn="fqdn=svr02.ipa.gmonlab.local,cn=computers,cn=accounts,dc=ipa,dc=gmo
nlab,dc=local"
```

In the Kerberos KDC log, a "Constrained Delegation" entry for the new domain member is recorded, as seen in the following example:

```
Jan 14 00:11:41 ipa01.ipa.gmonlab.local krb5kdc[5714](info): ...
CONSTRAINED-DELEGATION s4u-
client=host/svr02.ipa.gmonlab.local@IPA.GMONLAB.LOCAL
```

The Web UI error log contains an entry for the domain "join" which provides some additional details about the newly added system:

```
[Mon Jan 14 00:11:41.408760 2019] [:error] [pid 25236] ipa: INFO:
[xmlserver] admin@IPA.GMONLAB.LOCAL: join(u'svr02.ipa.gmonlab.local',
nshardwareplatform=u'x86_64', nsosversion=u'3.10.0-957.el7.x86_64',
version=u'2.51'): SUCCESS
```

4.2. User Activity

User activity involves events that occur when a user interacts with the FreeIPA domain environment. The presence of host-based access controls and other policies within FreeIPA can create a multitude of events. For this paper, user activity was constrained to include only those user events associated with user login, logout, password changes, and account lockouts. User activity is logged across multiple files on both the FreeIPA directory servers and the domain members (clients):

Log Name	System Location	File Location
Login Events	Domain Clients	/var/log/secure
Kerberos Administration Server	FreeIPA Servers	/var/log/kadmind.log
Kerberos Key Distribution Center (KDC)	FreeIPA Servers	/var/log/krb5kdc.log
FreeIPA Web UI Errors	FreeIPA Servers	/var/log/httpd/errors

4.2.1. User Logins

The Kerberos KDC log located on the FreeIPA directory servers record when a user attempts to log in to a domain connected machine. These log events are only generated when the client machine connects to the FreeIPA directory servers. If a user's credentials are cached on the client and the client is unable to talk to the directory servers, no log events are recorded in the KDC log. In the following log events, "NEEDED_PREAUTH" indicated a legitimate user account ID was provided, and the "AS_REQ" indicated a Kerberos ticket was issued for the user following a successful authentication:

```
Jan 13 22:27:35 ipa01.ipa.gmonlab.local krb5kdc[5714](info): AS_REQ (8
etypes {18 17 20 19 16 23 25 26}) 192.168.1.204: NEEDED_PREAUTH:
tuser5@IPA.GMONLAB.LOCAL for
krbtgt/IPA.GMONLAB.LOCAL@IPA.GMONLAB.LOCAL, Additional pre-
authentication required
```

```
Jan 13 22:27:35 ipa01.ipa.gmonlab.local krb5kdc[5714](info): closing
down fd 11

Jan 13 22:27:35 ipa01.ipa.gmonlab.local krb5kdc[5714](info): AS_REQ (8
etypes {18 17 20 19 16 23 25 26}) 192.168.1.204: ISSUE: authtime
1547418455, etypes {rep=18 tkt=18 ses=18}, tuser5@IPA.GMONLAB.LOCAL for
krbtgt/IPA.GMONLAB.LOCAL@IPA.GMONLAB.LOCAL
```

User login events are also stored on the client machine. On the CentOS 7 server and workstation domain members, SSH and local (TTY) logins were recorded locally in */var/log/secure*. The following is an example SSH login event:

```
Jan 13 23:56:54 wkst01 sshd[5754]: pam_unix(sshd:session): session
opened for user bgillam by (uid=0)
```

An example console login event is comprised of multiple log events:

```
Jan 13 23:58:23 wkst01 login: pam_sss(login:auth): authentication
success; logname=LOGIN uid=0 euid=0 tty=tty1 ruser= rhost= user=bgillam

Jan 13 23:58:23 wkst01 login: pam_unix(login:session): session opened
for user bgillam by LOGIN(uid=0)

Jan 13 23:58:23 wkst01 login: LOGIN ON tty1 BY bgillam
```

4.2.2. User Logouts

FreeIPA's components (389ds Directory Server, Web UI, and Kerberos daemons) do not log events associated with user logout activity. Domain member systems do record logout events locally in the */var/log/secure*. During testing, it was observed that domain members recorded enough information to differentiate between SSH and console (TTY) logouts. An example SSH logout event was:

```
Jan 13 23:45:22 wkst01 sshd[5477]: pam_unix(sshd:session): session
closed for user bgillam
```

An example console logout event was:

```
Jan 13 23:46:10 wkst01 login: pam_unix(login:session): session closed
for user bgillam
```

4.2.3. Password Changes

When a user logs into a workstation or server using SSH and changes their password with the *passwd* command, the Kerberos Administration Server (kadmind) on the FreeIPA directory servers log the password change request:

```
Jan 13 21:37:32 ipa01.ipa.gmonlab.local kadmind[5718](Notice): chpw
request from 192.168.1.204 for tuser5@IPA.GMONLAB.LOCAL: success
```

Errors that occur during the password change process are also recorded in the kadmind log. For example, if enforced password policies prevent users from quickly changing their password, the following error message is logged:

```
Jan 13 21:38:34 ipa01.ipa.gmonlab.local kadmind[5718](Notice): chpw
request from 192.168.1.204 for tuser5@IPA.GMONLAB.LOCAL: Current
password's minimum life has not expired
```

Users can also reset their password from the Web UI. When this occurs, a "passwd" event is recorded in the Web UI error log:

```
[Sun Jan 13 19:46:58.049050 2019] [:error] [pid 7720] ipa: INFO:
[jsonserver_session] tuser5@IPA.GMONLAB.LOCAL: passwd(u'tuser5',
u'********', None, version=u'2.229'): SUCCESS
```

4.2.4. Account Lockout

Account lockout occurs when a user has attempted to log in multiple times unsuccessfully. FreeIPA's default global password policy is set to 6 invalid attempts (Red Hat, 2018), but this value is dependent upon the organization's password policy.

When a valid username is provided but the account is locked out, LDAP SEARCH ("SRCH") operations are logged in the 389ds access log. However, the access logs do not provide any indication that the account is locked out. The "LOCKED_OUT" event is recorded in the Kerberos Key Distribution Center (KDC) log. The following is an example of this user account lockout event:

```
Jan 13 21:58:04 ipa01.ipa.gmonlab.local krb5kdc[5714](info): AS_REQ (8
etypes {18 17 20 19 16 23 25 26}) 192.168.1.204: LOCKED_OUT:
tuser5@IPA.GMONLAB.LOCAL for
```

```
krbtgt/IPA.GMONLAB.LOCAL@IPA.GMONLAB.LOCAL, Client's credentials have
been revoked
```

4.3. Abnormal/Malicious Activity

Abnormal and malicious behavior can take many forms within a centralized directory environment. Unauthorized access is the most common form, but for this paper, malicious activity was defined as unauthorized directory reconnaissance (i.e., directory exports), service account misuse, and password attacks (brute-force and password sprays). To effectively detect these malicious activities in a FreeIPA environment, logs from a single source was required:

Log Name	File Location
389ds Access	/var/log/dirsrv/slapd-IPA-<REALM>/access

4.3.1. Directory Reconnaissance

An example of directory reconnaissance is using Bloodhound in an Active Directory environment. Access to the centralized directory by any legitimate directory user allows an attacker to acquire details associated with all user accounts, groups, and group membership. While the attacker may not have access to pull password hashes, the directory still provides a wealth of targetable information. In FreeIPA, directory recon can take the form of a python script or tool (e.g., *ldapsearch*) issuing 1 or more all-encompassing queries.

To determine what a long running directory recon query looks like in the FreeIPA logs, 4000 test accounts were added to the IPA.GMONLAB.LOCAL domain. The following ldapsearch query was used to simulate an attacker dumping user accounts from the directory:

```
ldapsearch -x -b "cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local" -s
one -E \!pr=5000 -E \!sss=uid/givenName/sn "(objectclass=*)"
```

As shown below, the 389ds access log recorded the TLS connection created by ldapsearch as well as the search parameters and metadata on the search results:

```
[14/Jan/2019:02:16:56.141741833 +0000] conn=13254 fd=110 slot=110 SSL
connection from 192.168.1.206 to 192.168.1.201
[14/Jan/2019:02:16:56.223414277 +0000] conn=13254 TLS1.2 256-bit AES-
GCM
...
[14/Jan/2019:02:16:56.224072538 +0000] conn=13254 op=1 SRCH
base="cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local" scope=1
filter="(objectClass=*)" attrs=ALL
[14/Jan/2019:02:16:56.224327331 +0000] conn=13254 op=1 SORT uid
givenName sn (4010)
[14/Jan/2019:02:17:00.260490624 +0000] conn=13254 op=1 RESULT err=0
tag=101 nentries=4005 etime=4.0036611977 notes=U,P pr_idx=0 pr_cookie=-
1
```

This search activity is abnormal compared to normal directory usage within the test environment specifically because the LDAP SEARCH operation returned over 4000 entries (*nentries* attribute) and took just over 4 seconds to complete (*etime* attribute). In the test environment, most LDAP SEARCH operations returned less than 30 entries and completed in less than 1 second (~0.199 seconds). "Normal" in the controlled test environment was determined by observing the distribution of unique *nentries* and *etime* values prior to conducting "malicious" activity. The distribution was acquired using:

```
[root@ipa01 log]# cat /var/log/dirsrv/slapd-IPA-GMONLAB-LOCAL/access |
grep RESULT | cut -d" " -f8 | sort | uniq -c
[root@ipa01 log]# cat /var/log/dirsrv/slapd-IPA-GMONLAB-LOCAL/access |
grep RESULT | cut -d" " -f9 | sort -u
```

Normal patterns will differ between enterprise environments depending on the directory configuration. Defenders should consider long running queries that return abnormally large amounts of data to be suspect.

4.3.2. Service Account Misuse

Service accounts are used in enterprise environments wherever systems or applications need to perform lookups against the directory. Examples where services

accounts are used include on printers—to lookup email addresses for internal "scan to email" functionality—and in applications/systems that use LDAP for single sign-on (e.g. GitHub Enterprise Server, JIRA Service Desk, VPN servers.) In these situations, service accounts typically query group memberships and return results about a single user or group. The following is a real-life example of Moodle—a learning management platform—using a service account to perform an initial LDAP query during user login:

```
[14/Jan/2019:17:37:26.188936019 +0000] conn=13528 fd=133 slot=133
connection from 192.168.1.213 to 192.168.1.201

[14/Jan/2019:17:37:26.189387758 +0000] conn=13528 op=0 BIND
dn="uid=svcmoodle,cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local"
method=128 version=3

[14/Jan/2019:17:37:26.190355631 +0000] conn=13528 op=0 RESULT err=0
tag=97 nentries=0 etime=0.0001267199
dn="uid=svcmoodle,cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local"

[14/Jan/2019:17:37:26.191105463 +0000] conn=13528 op=1 SRCH
base="cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local" scope=1
filter="(&(objectClass=*)(uid=tuser87))" attrs="uid"

[14/Jan/2019:17:37:26.191651525 +0000] conn=13528 op=1 RESULT err=0
tag=101 nentries=1 etime=0.0000728723 notes=U

[14/Jan/2019:17:37:26.192094990 +0000] conn=13528 op=2 BIND
dn="uid=tuser87,cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local"
method=128 version=3

[14/Jan/2019:17:37:26.192700121 +0000] conn=13528 op=2 RESULT err=0
tag=97 nentries=0 etime=0.0000682841
dn="uid=tuser87,cn=users,cn=accounts,dc=ipa,dc=gmonlab,dc=local"

[14/Jan/2019:17:37:26.193065547 +0000] conn=13528 op=3 UNBIND

[14/Jan/2019:17:37:26.193074875 +0000] conn=13528 op=3 fd=133 closed -
U1
```

In the above example, user "tuser87" attempts to login to the Moodle web application that is hosted on 192.168.1.213. Moodle connected to the FreeIPA directory using the service account "svcmoodle" and performed a search for the user id "tuser87". Once Moodle verified the user id is in the correct group, Moodle then performed an

LDAP BIND operation using the "tuser87" account within the same LDAP connection (13528).

Understanding what is typical for a service account is a prerequisite to determining if the account is compromised. Examples of unusual service account activity include logins from unexpected locations and LDAP queries that go beyond the service accounts expected scope.

4.3.3. Password Attacks

Credential compromise is a common attack vector in enterprise environments. Two techniques attackers used to discover account passwords are password brute-force guessing and password spraying.

Password brute-force guessing occurs when an attacker tries multiple passwords against a single account (Javed, 2016). A brute-force was conducted using *ncrack* to determine what a password brute-force attack looks like in the FreeIPA logs. The FreeIPA directory server access logs recorded LDAP SEARCH operations associated this brute-force activity:

```
[14/Jan/2019:20:17:54.159743277 +0000] conn=13569 fd=110 slot=110
connection from 192.168.1.203 to 192.168.1.201
...
[14/Jan/2019:20:17:54.212188176 +0000] conn=13569 op=4 SRCH
base="cn=accounts,dc=ipa,dc=gmonlab,dc=local" scope=2
filter="(&(uid=user10)(objectClass=posixAccount)(uid=*)(&(uidNumber=*)(
!(uidNumber=0))))" attrs="objectClass uid userPassword uidNumber
gidNumber gecos homeDirectory loginShell krbPrincipalName cn memberOf
ipaUniqueID ipaNTSecurityIdentifier modifyTimestamp entryusn
shadowLastChange shadowMin shadowMax shadowWarning shadowInactive
shadowExpire shadowFlag krbLastPwdChange krbPasswordExpiration
pwdattribute authorizedService accountexpires useraccountcontrol
nsAccountLock host logindisabled loginexpiretiontime
loginallowedtimemap ipaSshPubKey ipaUserAuthType usercertificate;binary
mail"
...
[14/Jan/2019:20:26:05.088204101 +0000] conn=13569 op=67 SRCH
```

```
base="cn=accounts,dc=ipa,dc=gmonlab,dc=local" scope=2
filter="(&(uid=user10)(objectClass=posixAccount)(uid=*)(&(uidNumber=*)(
!(uidNumber=0))))" attrs="objectClass uid userPassword uidNumber
gidNumber gecos homeDirectory loginShell krbPrincipalName cn memberOf
ipaUniqueID ipaNTSecurityIdentifier modifyTimestamp entryusn
shadowLastChange shadowMin shadowMax shadowWarning shadowInactive
shadowExpire shadowFlag krbLastPwdChange krbPasswordExpiration
pwdattribute authorizedService accountexpires useraccountcontrol
nsAccountLock host logindisabled loginexpirationtime
loginallowedtimemap ipaSshPubKey ipaUserAuthType usercertificate;binary
mail"
```

An important artifact to note is that in the sample log, all authentication attempts occurred over the same LDAP connection (13569) with the FreeIPA directory server with the operation number ("op") increasing.

A password spray attack involves an attacker testing either a small list of the most common passwords or a single password against multiple user accounts (Wilkin, 2017). This attack works well against organizations that reset user passwords to a known value that is specific to the company (e.g., P@ssword123) or time of year (e.g., Spring2019!). Directory server access logs recorded LDAP SEARCH operations associated with the password spraying:

```
[15/Jan/2019:00:32:55.005139793 +0000] conn=13620 fd=110 slot=110
connection from 192.168.1.203 to 192.168.1.201
...
[15/Jan/2019:00:42:45.622681612 +0000] conn=13620 op=223 SRCH
base="cn=accounts,dc=ipa,dc=gmonlab,dc=local" scope=2
filter="(&(uid=user403)(objectClass=posixAccount)(uid=*)(&(uidNumber=*)
(!(uidNumber=0))))" attrs="objectClass uid userPassword uidNumber
gidNumber gecos homeDirectory loginShell krbPrincipalName cn memberOf
ipaUniqueID ipaNTSecurityIdentifier modifyTimestamp entryusn
shadowLastChange shadowMin shadowMax shadowWarning shadowInactive
shadowExpire shadowFlag krbLastPwdChange krbPasswordExpiration
pwdattribute authorizedService accountexpires useraccountcontrol
nsAccountLock host logindisabled loginexpirationtime
loginallowedtimemap ipaSshPubKey ipaUserAuthType usercertificate;binary
```

```
mail"
...
[15/Jan/2019:00:43:56.505645727 +0000] conn=13620 op=527 SRCH
base="cn=accounts,dc=ipa,dc=gmonlab,dc=local" scope=2
filter="(&(uid=user989)(objectClass=posixAccount)(uid=*)(&(uidNumber=*)
(!(uidNumber=0))))" attrs="objectClass uid userPassword uidNumber
gidNumber gecos homeDirectory loginShell krbPrincipalName cn memberOf
ipaUniqueID ipaNTSecurityIdentifier modifyTimestamp entryusn
shadowLastChange shadowMin shadowMax shadowWarning shadowInactive
shadowExpire shadowFlag krbLastPwdChange krbPasswordExpiration
pwdattribute authorizedService accountexpires useraccountcontrol
nsAccountLock host logindisabled loginexpirationtime
loginallowedtimemap ipaSshPubKey ipaUserAuthType usercertificate;binary
mail"
```

Activity associated with password spraying attacks in the 389ds access log exhibited a similar pattern to the password brute-force guessing attack: a single LDAP connection executed hundreds of operations. The significant difference was that hundreds of usernames appeared in the user id filter (e.g. user 403, user 404) as the attacker iterated over the user accounts.

In Active Directory environments, detecting activity associated with password attacks requires correlation and analysis of Windows Event ID 4625 (Milford, 2017). In FreeIPA environments, an additional analytical effort is required to parse multiple lines in the directory service logs as a single malicious event. Long tail analysis against LDAP connections—using the following command and looking specifically for SEARCH operations that use a uid filter—causes both password attack activities to standout:

```
[root@ipa01 log]# cat /var/log/dirsrv/slapd-IPA-GMONLAB-LOCAL/access |
grep SRCH | grep "filter=\"(&(uid=" | cut -d" " -f3 | uniq -c
```

5. Conclusions

The focus of existing Continuous Security Monitoring resources on Active Directory-specific techniques has resulted in a significant gap for non-AD directory services. Knowing what is logged by in a FreeIPA domain with regards to administrative, user, and known malicious activity is paramount to successfully implementing a continuous security monitoring solution. FreeIPA's default logging configuration captures these events in a format that allows defenders to extract value quickly. In most cases, the same knowledge and observations apply to OpenLDAP-based directory services with minimal modifications.

5.1. Future Research

The observations presented in this research are a point in time snapshot using FreeIPA v4.6.4 on CentOS 7. The techniques that require FreeIPA Web UI error logs are not useful in OpenLDAP-based environments. This presents an opportunity for future research on more generalized CSM methods that can be applied across OpenLDAP-based directory servers.

References

Atkisson, B. (2016, April 29). Red Hat Identity Manager: Part 1 - Overview and Getting started - RHD Blog. Retrieved February 6, 2019, from https://developers.redhat.com/blog/2016/04/29/red-hat-identity-manager-part-1-overview-and-getting-started/

Delgado, P. (2018, August 20). Monitoring Active Directory with ELK. Retrieved November 10, 2018, from https://www.syspanda.com/index.php/2018/05/03/monitoring-active-directory-elk/

FireEye. (2018, April 4). M-Trends 2018. Retrieved from https://www.fireeye.com/content/dam/collateral/en/mtrends-2018.pdf

Fredriksson, T. (2010). Implementing LDAPv3: OpenLDAP, Kerberos V and glue code for distributed data. 2nd ed.

Gartner, Inc. (2017, March 14). Gartner Says Detection and Response is Top Security Priority for Organizations in 2017. Retrieved from https://www.gartner.com/en/newsroom/press-releases/2017-03-14-gartner-says-detection-and-response-is-top-security-priority-for-organizations-in-2017

Harrison, R. (2006). Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms (No. RFC 4513).

Javed, M. (2016). Detecting credential compromise in enterprise networks (Doctoral dissertation). Retrieved from https://escholarship.org/uc/item/0sh3b0tw

Yuriev, L. (2018, December 17). ReOpenLDAP. Retrieved from https://github.com/leo-yuriev/ReOpenLDAP

Metcalf, S. (2015, July 15). It's All About Trust – Forging Kerberos Trust Tickets to Spoof Access across Active Directory Trusts. Retrieved from https://adsecurity.org/?p=1588

Metcalf, S. (2018, October 19). Attack Defense & Detection. Retrieved from https://adsecurity.org/?page_id=4031

Microsoft. (2017, May 30). Monitoring Active Directory for Signs of Compromise. Retrieved December 8, 2018, from https://docs.microsoft.com/en-us/windows-

server/identity/ad-ds/plan/security-best-practices/monitoring-active-directory-for-signs-of-compromise

Milford, A. (2017, January 4). Auditing Remote Desktop Services Logon Failures (Part 1) - PureRDS. Retrieved from http://purerds.org/remote-desktop-security/auditing-remote-desktop-services-logon-failures-1/

Newton, H. (2002). Newton's Telecom Dictionary (18th ed.). CMP Books.

OpenLDAP Foundation. (2015, July 1). OpenLDAP. Retrieved from http://www.openldap.org/

Oracle. (n.d.). Oracle Internet Directory. Retrieved from https://www.oracle.com/technetwork/middleware/id-mgmt/overview/index-082035.html

Red Hat. (n.d.). 389 Directory Server. Retrieved from https://directory.fedoraproject.org/index.html

Red Hat. (2018, October 29). *Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy Guide*. Retrieved from https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html-single/Linux_Domain_Identity_Authentication_and_Policy_Guide/

Robbins, A. (2016, August 29). Introducing BloodHound. Retrieved October 20, 2018, from https://wald0.com/?p=68

Rodriguez, R. (2018, April 9). Welcome to HELK!: Enabling Advanced Analytics Capabilities. Retrieved November 10, 2018, from https://cyberwardog.blogspot.com/2018/04/welcome-to-helk-enabling-advanced_9.html

Schiffer, P. (2017, October 20). ipa-log-config. Retrieved December 29, 2018, from https://github.com/pschiffe/ipa-log-config

Schiffer, P. (2015). rsyslog-elasticsearch-kibana. Retrieved December 29, 2018, from https://github.com/pschiffe/rsyslog-elasticsearch-kibana

Sermersheim, J. (2006). Lightweight directory access protocol (LDAP): The protocol (No. RFC 4511).

Blair G

Wilkin, J. (2017, December 1). Simplifying Password Spraying. Retrieved from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/simplifying-password-spraying/

Blair Gillam;blair@detnstudios.com

Appendix A

The Packer configuration files, CentOS 7 kickstarts, and test scripts used to generate the virtual lab and create the log events are available at

https://github.com/littleairmada/gmon-ipa-lab