2.3 Configure Time Synchronization

It is recommended that systems be configured to synchronize their time using a service such as `systemd-timesyncd`, or `chrony`.

Virtual systems may be configured to receive their time synchronization from their host system.

The host system must be configured to synchronize its time from an authoritative source to be considered compliant with this section.

Any "physical" clock present on a system should be synchronized from an authoritative time source.

Only one time synchronization method should be in use on the system

Notes: Only the section related to the time synchronization method in use on the system should be followed, all other time synchronization recommendations should be skipped

2.3.1 Ensure time synchronization is in use

It is recommended that physical systems and virtual guests lacking direct access to the physical host's clock be configured to synchronize their time using a service such as `systemd-timesyncd`, or `chrony`.

2.3.1.1 Ensure a single time synchronization daemon is in use (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

System time should be synchronized between all systems in an environment. This is typically done by establishing an authoritative time server or set of servers and having all systems synchronize their clocks to them.

Note:

- **On virtual systems where host based time synchronization is available consult your virtualization software documentation and verify that host based synchronization is in use and follows local site policy. In this scenario, this section should be skipped**
- Only **one** time synchronization method should be in use on the system. Configuring multiple time synchronization methods could lead to unexpected or unreliable results

Rationale:

Time synchronization is important to support time sensitive security mechanisms and ensures log files have consistent time records across the enterprise, which aids in forensic investigations.

Audit:

On physical systems, and virtual systems where host based time synchronization is not available.
One of the two time synchronization daemons should be available; **`chrony`** or **`systemd-timesyncd`**
Run the following script to verify that a single time synchronization daemon is available on the system:

```
#!/usr/bin/env bash

{
    l_output="" l_output2=""
    service_not_enabled_chk()
    {
        l_out2=""
        if systemctl is-enabled "$l_service_name" 2>/dev/null | grep -q 'enabled'; then
            l_out2="$l_out2\n  - Daemon: \"$l_service_name\" is enabled on the system"
        fi
        if systemctl is-active "$l_service_name" 2>/dev/null | grep -q '^active'; then
            l_out2="$l_out2\n  - Daemon: \"$l_service_name\" is active on the system"
        fi
    }
    l_service_name="systemd-timesyncd.service" # Check systemd-timesyncd daemon
    service_not_enabled_chk
    if [ -n "$l_out2" ]; then
        l_timesyncd="y"
        l_out_tsd="$l_out2"
    else
        l_timesyncd="n"
        l_out_tsd="\n  - Daemon: \"$l_service_name\" is not enabled and not active on the system"
    fi
    l_service_name="chrony.service" # Check chrony
    service_not_enabled_chk
    if [ -n "$l_out2" ]; then
        l_chrony="y"
        l_out_chrony="$l_out2"
    else
        l_chrony="n"
        l_out_chrony="\n  - Daemon: \"$l_service_name\" is not enabled and not active on the
system"
    fi
    l_status="$l_timesyncd$l_chrony"
    case "$l_status" in
        yy)
            l_output2=" - More than one time sync daemon is in use on the
system$l_out_tsd$l_out_chrony"
            ;;
        nn)
            l_output2=" - No time sync daemon is in use on the system$l_out_tsd$l_out_chrony"
            ;;
        yn|ny)
            l_output=" - Only one time sync daemon is in use on the
system$l_out_tsd$l_out_chrony"
            ;;
        *)
            l_output2=" - Unable to determine time sync daemon(s) status"
            ;;
    esac

    if [ -z "$l_output2" ]; then
        echo -e "\n- Audit Result:\n  ** PASS **\n$l_output\n"
    else
        echo -e "\n- Audit Result:\n  ** FAIL **\n - * Reasons for audit failure *
:\n$l_output2\n"
    fi
}
```

Note: Follow the guidance in the subsection for the time synchronization daemon available on the system and skip the other time synchronization daemon subsection.

Remediation:

On physical systems, and virtual systems where host based time synchronization is not available.
Select **one** of the two time synchronization daemons; **`chrony (1)`** or **`systemd-timesyncd (2)`** and following the remediation procedure for the selected daemon.
Note: enabling more than one synchronization daemon could lead to unexpected or unreliable results:

1. **`chrony`**

Run the following command to install `chrony`:

```
# apt install chrony
```

Run the following commands to stop and mask the `systemd-timesyncd` daemon:

```
# systemctl stop systemd-timesyncd.service

# systemctl mask systemd-timesyncd.service
```

Note:

- Subsection: ***Configure chrony*** should be followed
- Subsection: ***Configure systemd-timesyncd*** should be skipped

2. **`systemd-timesyncd`**

Run the following command to remove the chrony package:

```
# apt purge chrony
# apt autoremove chrony
```

Note:

- Subsection: ***Configure systemd-timesyncd*** should be followed
- Subsection: ***Configure chrony*** should be skipped

References:

1. NIST SP 800-53 Rev. 5: AU-3, AU-12

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0005	

2.3.2 Configure systemd-timesyncd

`systemd-timesyncd` is a daemon that has been added for synchronizing the system clock across the network. It implements an SNTP client. In contrast to NTP implementations such as chrony or the NTP reference server this only implements a client side, and does not bother with the full NTP complexity, focusing only on querying time from one remote server and synchronizing the local clock to it. The daemon runs with minimal privileges, and has been hooked up with networkd to only operate when network connectivity is available. The daemon saves the current clock to disk every time a new NTP sync has been acquired, and uses this to possibly correct the system clock early at bootup, in order to accommodate for systems that lack an RTC such as the Raspberry Pi and embedded devices, and make sure that time monotonically progresses on these systems, even if it is not always correct. To make use of this daemon a new system user and group "systemd-timesync" needs to be created on installation of systemd.

The default configuration is set during compilation, so configuration is only needed when it is necessary to deviate from those defaults. Initially, the main configuration file in /etc/systemd/ contains commented out entries showing the defaults as a guide to the administrator. Local overrides can be created by editing this file or by creating drop-ins, as described below. Using drop-ins for local configuration is recommended over modifications to the main configuration file.

In addition to the "main" configuration file, drop-in configuration snippets are read from `/usr/lib/systemd/*.conf.d/`, `/usr/local/lib/systemd/*.conf.d/`, and `/etc/systemd/*.conf.d/`. Those drop-ins have higher precedence and override the main configuration file. Files in the *.conf.d/ configuration subdirectories are sorted by their filename in lexicographic order, regardless of in which of the subdirectories they reside. When multiple files specify the same option, for options which accept just a single value, the entry in the file sorted last takes precedence, and for options which accept a list of values, entries are collected as they occur in the sorted files.

When packages need to customize the configuration, they can install drop-ins under /usr/. Files in /etc/ are reserved for the local administrator, who may use this logic to override the configuration files installed by vendor packages. Drop-ins have to be used to override package drop-ins, since the main configuration file has lower precedence. It is recommended to prefix all filenames in those subdirectories with a two-digit number and a dash, to simplify the ordering of the files.

To disable a configuration file supplied by the vendor, the recommended way is to place a symlink to /dev/null in the configuration directory in /etc/, with the same filename as the vendor configuration file.

Note:

- The recommendations in this section only apply if `timesyncd` is in use on the system
- The `systemd-timesyncd` service specifically implements only SNTP.
  - This minimalistic service will set the system clock for large offsets or slowly adjust it for smaller deltas
  - More complex use cases are not covered by `systemd-timesyncd`
- **If `chrony` is used, `systemd-timesyncd` should be stopped and masked, and this section skipped**
- **One, and only one, time synchronization method should be in use on the system**

2.3.2.1 Ensure systemd-timesyncd configured with authorized timeserver (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

`NTP=`

- A space-separated list of NTP server host names or IP addresses. During runtime this list is combined with any per-interface NTP servers acquired from systemd-networkd.service(8). systemd-timesyncd will contact all configured system or per-interface servers in turn, until one responds. When the empty string is assigned, the list of NTP servers is reset, and all prior assignments will have no effect. This setting defaults to an empty list.

`FallbackNTP=`

- A space-separated list of NTP server host names or IP addresses to be used as the fallback NTP servers. Any per-interface NTP servers obtained from systemd-networkd.service(8) take precedence over this setting, as do any servers set via NTP= above. This setting is hence only relevant if no other NTP server information is known. When the empty string is assigned, the list of NTP servers is reset, and all prior assignments will have no effect. If this option is not given, a compiled-in list of NTP servers is used.

Rationale:

Time synchronization is important to support time sensitive security mechanisms and to ensure log files have consistent time records across the enterprise to aid in forensic investigations

Audit:

Run the following command to verify the `NTP` **and/or** `FallbackNTP` option is set to local site approved authoritative time server(s):

```
#!/usr/bin/env bash

{
   a_output=(); a_output2=(); a_parlist=("NTP=[^#\n\r]+" "FallbackNTP=[^#\n\r]+")
   l_systemd_config_file="/etc/systemd/timesyncd.conf" # Main systemd configuration file
   f_config_file_parameter_chk()
   {
      unset A_out; declare -A A_out # Check config file(s) setting
      while read -r l_out; do
         if [ -n "$l_out" ]; then
            if [[ $l_out =~ ^\s*# ]]; then
               l_file="${l_out//# /}"
            else
               l_systemd_parameter="$(awk -F= '{print $1}' <<< "$l_out" | xargs)"
               grep -Piq -- "^\h*$l_systemd_parameter_name\b" <<< "$l_systemd_parameter" &&
A_out+=(["$l_systemd_parameter"]="$l_file")
            fi
         fi
      done < <("$l_systemdanalyze" cat-config "$l_systemd_config_file" | grep -Pio
'^\h*([^#\n\r]+|#\h*\/[^#\n\r\h]+\.conf\b)')
      if (( ${#A_out[@]} > 0 )); then # Assess output from files and generate output
         while IFS="=" read -r l_systemd_file_parameter_name l_systemd_file_parameter_value; do
            l_systemd_file_parameter_name="${l_systemd_file_parameter_name// /}"
            l_systemd_file_parameter_value="${l_systemd_file_parameter_value// /}"
            if grep -Piq "\b$l_systemd_parameter_value\b" <<< "$l_systemd_file_parameter_value";
then
               a_output+=(" - \"$l_systemd_parameter_name\" is correctly set to
\"$l_systemd_file_parameter_value\"" \
               "    in \"$(printf '%s' "${A_out[@]}")\"")
            else
               a_output2+=(" - \"$l_systemd_parameter_name\" is incorrectly set to
\"$l_systemd_file_parameter_value\"" \
               "    in \"$(printf '%s' "${A_out[@]}")\" and should have a value matching:
\"$l_value_out\"")
            fi
         done < <(grep -Pio -- "^\h*$l_systemd_parameter_name\h*=\h*\H+" "${A_out[@]}")
      else
         a_output2+=(" - \"$l_systemd_parameter_name\" is not set in an included file" \
         "   *** Note: \"$l_systemd_parameter_name\" May be set in a file that's ignored by load
procedure ***")
      fi
   }
   l_systemdanalyze="$(readlink -f /bin/systemd-analyze)"
   while IFS="=" read -r l_systemd_parameter_name l_systemd_parameter_value; do # Assess and
check parameters
      l_systemd_parameter_name="${l_systemd_parameter_name// /}";
l_systemd_parameter_value="${l_systemd_parameter_value// /}"
      l_value_out="${l_systemd_parameter_value//-/ through }"; l_value_out="${l_value_out//|/ or
}"
      l_value_out="$(tr -d '(){}' <<< "$l_value_out")"
      f_config_file_parameter_chk
   done < <(printf '%s\n' "${a_parlist[@]}")
   if [ "${#a_output2[@]}" -le 0 ]; then
      printf '%s\n' "" "- Audit Result:" "  ** PASS **" "${a_output[@]}" ""
   else
      printf '%s\n' "" "- Audit Result:" "  ** FAIL **" " - Reason(s) for audit failure:"
"${a_output2[@]}"
      [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:" "${a_output[@]}" ""
   fi
}
```

Example output:

```
- Audit Result:
  ** PASS **
 - "NTP" is correctly set to "time.nist.gov"
   in "/etc/systemd/timesyncd.conf.d/60-timesyncd.conf"
 - "FallbackNTP" is correctly set to "time-a-g.nist.gov"
   in "/etc/systemd/timesyncd.conf.d/60-timesyncd.conf"
```

Note: Please ensure the output for `NTP` and/or `FallbackNTP` is in accordance with local site policy. The timeservers in the example output are provided as an example of possible timeservers and they may not follow local site policy.

Remediation:

Set `NTP` and/or `FallbackNPT` parameters to local site approved authoritative time server(s) in `/etc/systemd/timesyncd.conf` or a file in `/etc/systemd/timesyncd.conf.d/` ending in `.conf` in the `[Time]` section:
Example file:

```
[Time]
NTP=time.nist.gov # Uses the generic name for NIST's time servers
FallbackNTP=time-a-g.nist.gov time-b-g.nist.gov time-c-g.nist.gov # Space
separated list of NIST time servers
```

Example script to create systemd drop-in configuration file:

```
#!/usr/bin/env bash

{
   a_settings=("NTP=time.nist.gov" "FallbackNTP=time-a-g.nist.gov time-b-
g.nist.gov time-c-g.nist.gov")
   [ ! -d /etc/systemd/timesyncd.conf.d/ ] && mkdir
/etc/systemd/timesyncd.conf.d/
   if grep -Psq -- '^\h*\[Time\]' /etc/systemd/timesyncd.conf.d/60-
timesyncd.conf; then
      printf '%s\n' "" "${a_settings[@]}" >>
/etc/systemd/timesyncd.conf.d/60-timesyncd.conf
   else
      printf '%s\n' "" "[Time]" "${a_settings[@]}" >>
/etc/systemd/timesyncd.conf.d/60-timesyncd.conf
   fi
}
```

Note: If this setting appears in a canonically later file, or later in the same file, the setting will be overwritten
Run to following command to update the parameters in the service:

```
# systemctl reload-or-restart systemd-journald
```

Default Value:

#NTP=

#FallbackNTP=

References:

1. https://www.freedesktop.org/software/systemd/man/timesyncd.conf.html
2. https://tf.nist.gov/tf-cgi/servers.cgi
3. NIST SP 800-53 Rev. 5: AU-7, AU-8

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0002	M1022

2.3.2.2 Ensure systemd-timesyncd is enabled and running (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

systemd-timesyncd is a daemon that has been added for synchronizing the system clock across the network

Rationale:

systemd-timesyncd needs to be enabled and running in order to synchronize the system to a timeserver.

Time synchronization is important to support time sensitive security mechanisms and to ensure log files have consistent time records across the enterprise to aid in forensic investigations

Audit:

- IF - systemd-timesyncd is in use on the system, run the following commands:
Run the following command to verify that the `systemd-timesyncd` service is enabled:

```
# systemctl is-enabled systemd-timesyncd.service

enabled
```

Run the following command to verify that the `systemd-timesyncd` service is active:

```
# systemctl is-active systemd-timesyncd.service

active
```

Remediation:

- IF - `systemd-timesyncd` is in use on the system, run the following commands:
Run the following command to unmask `systemd-timesyncd.service`:

```
# systemctl unmask systemd-timesyncd.service
```

Run the following command to enable and start `systemd-timesyncd.service`:

```
# systemctl --now enable systemd-timesyncd.service
```

- OR -
If another time synchronization service is in use on the system, run the following command to stop and mask `systemd-timesyncd`:

```
# systemctl --now mask systemd-timesyncd.service
```

References:

1. NIST SP 800-53 Rev. 5: AU-7, AU-8

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0002	M1022

2.3.3 Configure chrony

`chrony` is a daemon which implements the Network Time Protocol (NTP) and is designed to synchronize system clocks across a variety of systems and use a source that is highly accurate.

`chrony` can be configured to be a client and/or a server.

More information on `chrony` can be found at: http://chrony.tuxfamily.org/.

Note:

- If `systemd-timesyncd` is being used, `chrony` should be removed and this section skipped
- Only one time synchronization method should be in use on the system

2.3.3.1 Ensure chrony is configured with authorized timeserver (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

- server
  - The server directive specifies an NTP server which can be used as a time source. The client-server relationship is strictly hierarchical: a client might synchronize its system time to that of the server, but the server's system time will never be influenced by that of a client.
  - This directive can be used multiple times to specify multiple servers.
  - The directive is immediately followed by either the name of the server, or its IP address.
- pool
  - The syntax of this directive is similar to that for the server directive, except that it is used to specify a pool of NTP servers rather than a single NTP server. The pool name is expected to resolve to multiple addresses which might change over time.
  - This directive can be used multiple times to specify multiple pools.
  - All options valid in the server directive can be used in this directive too.

Rationale:

Time synchronization is important to support time sensitive security mechanisms and to ensure log files have consistent time records across the enterprise to aid in forensic investigations

Audit:

- IF - `chrony` is in use on the system, run the following script to ensure `chrony` is configured with an authorized timeserver:

```
#!/usr/bin/env bash

{
   a_output=() a_output2=() a_config_files=("/etc/chrony/chrony.conf")
   l_include='(confdir|sourcedir)' l_parameter_name='(server|pool)'
l_parameter_value='.+'
   while IFS= read -r l_conf_loc; do
      l_dir="" l_ext=""
      if [ -d "$l_conf_loc" ]; then
         l_dir="$l_conf_loc" l_ext="*"
      elif  grep -Psq '\/\*\.([^#/\n\r]+)?\h*$' <<< "$l_conf_loc" || [ -f
"$(readlink -f "$l_conf_loc")" ]; then
         l_dir="$(dirname "$l_conf_loc")" l_ext="$(basename "$l_conf_loc")"
      fi
      if [[ -n "$l_dir" && -n "$l_ext" ]]; then
         while IFS= read -r -d $'\0' l_file_name; do
            [ -f "$(readlink -f "$l_file_name")" ] &&
a_config_files+=("$(readlink -f "$l_file_name")")
         done < <(find -L "$l_dir" -type f -name "$l_ext" -print0
2>/dev/null)
      fi
   done < <(awk '$1~/^\s*'"$l_include"'$/{print $2}' "${a_config_files[*]}"
2>/dev/null)
   for l_file in "${a_config_files[@]}"; do
      l_parameter_line="$(grep -Psi
'^\h*'"$l_parameter_name"'(\h+|\h*:\h*)'"$l_parameter_value"'\b' "$l_file")"
      [ -n "$l_parameter_line" ] && a_output+=("  - Parameter: \"$(tr -d '()'
<<< ${l_parameter_name//|/ or })\"" \
      "    Exists in the file: \"$l_file\" as:" "$l_parameter_line")
   done
   [ "${#a_output[@]}" -le "0" ] && a_output2+=("  - Parameter: \"$(tr -d
'()' <<< ${l_parameter_name//|/ or })\"" \
   "    Does not exist in the chrony configuration")
   if [ "${#a_output2[@]}" -le 0 ]; then
      printf '%s\n' "" "- Audit Result:" "  ** PASS **" "${a_output[@]}" ""
   else
      printf '%s\n' "" "- Audit Result:" "  ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
   fi
}
```

Remediation:

Edit `/etc/chrony/chrony.conf` or a file ending in `.sources` in `/etc/chrony/sources.d/` and add or edit server or pool lines as appropriate according to local site policy:
Edit the `Chrony` configuration and add or edit the server and/or pool lines returned by the Audit Procedure as appropriate according to local site policy

```
<[server|pool]> <[remote-server|remote-pool]>
```

Example script to add a drop-in configuration for the `pool` directive:

```
#!/usr/bin/env bash

{
   [ ! -d "/etc/chrony/sources.d/" ] && mkdir /etc/chrony/sources.d/
   printf '%s\n' "" "#The maxsources option is unique to the pool directive"
\
   "pool time.nist.gov iburst maxsources 4" >> /etc/chrony/sources.d/60-
sources.sources
   chronyc reload sources &>/dev/null
}
```

Example script to add a drop-in configuration for the `server` directive:

```
#!/usr/bin/env bash

{
   [ ! -d "/etc/chrony/sources.d/" ] && mkdir /etc/chrony/sources.d/
   printf '%s\n' "" "server time-a-g.nist.gov iburst" "server 132.163.97.3
iburst" \
   "server time-d-b.nist.gov iburst" >> /etc/chrony/sources.d/60-
sources.sources
   chronyc reload sources &>/dev/null
}
```

Run the following command to reload the `chronyd` config:

```
# systemctl reload-or-restart chronyd
```

References:

1. chrony.conf(5) Manual Page
2. https://tf.nist.gov/tf-cgi/servers.cgi
3. NIST SP 800-53 Rev. 5: AU-3, AU-12

Additional Information:

If pool and/or server directive(s) are set in a sources file in `/etc/chrony/sources.d`, the line:

```
sourcedir /etc/chrony/sources.d
```

must be present in `/etc/chrony/chrony.conf`

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0002	M1022

2.3.3.2 Ensure chrony is running as user _chrony (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `chrony` package is installed with a dedicated user account `_chrony`. This account is granted the access required by the `chronyd` service

Rationale:

The `chronyd` service should run with only the required privlidges

Audit:

- IF - `chrony` is in use on the system, run the following command to verify the `chronyd` service is being run as the `_chrony` user:

```
# ps -ef | awk '(/[c]hronyd/ && $1!="_chrony") { print $1 }'
```

Nothing should be returned

Remediation:

Add or edit the `user` line to `/etc/chrony/chrony.conf` or a file ending in `.conf` in `/etc/chrony/conf.d/`:

```
user _chrony
```

- OR -
If another time synchronization service is in use on the system, run the following command to remove `chrony` from the system:

```
# apt purge chrony
# apt autoremove chrony
```

Default Value:

user _chrony

References:

1. NIST SP 800-53 Rev. 5: AU-8

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0002	M1022