Intel® Unite™ Solution 3.0 and Protected Guest Access

Security Development Summary
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>Audience</td>
<td>4</td>
</tr>
<tr>
<td>Glossary</td>
<td>4</td>
</tr>
<tr>
<td>Security and Privacy objective based approach</td>
<td>5</td>
</tr>
<tr>
<td>Secure Development Lifecycle</td>
<td>6</td>
</tr>
<tr>
<td>Architecture and Design</td>
<td>6</td>
</tr>
<tr>
<td>Implementation</td>
<td>6</td>
</tr>
<tr>
<td>Validation</td>
<td>6</td>
</tr>
<tr>
<td>Release</td>
<td>6</td>
</tr>
<tr>
<td>Product Architecture Overview</td>
<td>7</td>
</tr>
<tr>
<td>Intel® Unite™ Solution Architecture Overview</td>
<td>7</td>
</tr>
<tr>
<td>Meeting Lock</td>
<td>8</td>
</tr>
<tr>
<td>Scheduled Meeting</td>
<td>9</td>
</tr>
<tr>
<td>WebRTC support for Video with Audio Capabilities</td>
<td>9</td>
</tr>
<tr>
<td>Intel® Unite™ solution Protected Guest Access Architecture Overview</td>
<td>9</td>
</tr>
<tr>
<td>Security Objectives</td>
<td>13</td>
</tr>
<tr>
<td>Product Security Objectives</td>
<td>13</td>
</tr>
<tr>
<td>Architecture Security Objectives</td>
<td>17</td>
</tr>
<tr>
<td>Addendum</td>
<td>18</td>
</tr>
<tr>
<td>Changes to the system environment</td>
<td>18</td>
</tr>
</tbody>
</table>

June 2016
Introduction

This document is being provided in response to the recent inquiry made to Intel requesting threat modelling details and security assurance practices for several projects. As communicated in the past, security and privacy are components of Intel’s culture and those components assist in the design, development, and delivery of our products and services. Intel offers a widely varied set of products and services, and through these, we strive to meet the security and privacy needs of our global customers. In addition, Intel has well established policies regarding the type and level of security information it shares with external parties. This approach is designed to balance customer needs with Intel's need to maintain the confidentiality of our classified information.

It is unusual to share additional details. The threat modeling details and security assurance practices provided herein are applicable to specific projects, namely the Intel® Unite™ solution, and should not be attributed to additional products or projects.

Audience

This document is designed for use by IT professionals and for other audiences that need additional details in regards to the security and privacy practices within the architecture and secure development lifecycle for the Intel Unite solution.

Glossary

Asset: Something of value intrinsic to your ingredient that requires protection
Actor: An agent with a certain privilege and capability that can act with a malicious intent
Attack point: An interface that an actor uses, or the means through which an actor gains access to an asset
Technique: A unique way of violating a protection defined, that an actor may use with the granted capabilities
Protection required: Includes Confidentiality, Integrity, and Availability
Mitigation strategy: A way employed to mitigate a threat
MITM: Man in the Middle Attack
WebRTC: Open source real-time communication library
SDL: Secure Development Lifecycle
Security and Privacy objective based approach

Below is the introduction to the approach that we have taken.

For these purposes, a threat is uniquely expressed using the below attributes:

- Asset
- Actor/Attacker
- Technique
- Attack point
- Protection required
- Mitigation strategy

Hence, a threat can be uniquely expressed as: An (actor) uses a specific (technique) through an (attack point) violating a (protection required) on an (asset) that we have defined. If a mitigation strategy is expressed, this implies that a particular threat is intended to be mitigated, this becomes a security objective for the project. However if the asset associated is privacy related, this becomes a privacy objective. Similarly if a threat is not intended to be mitigated this becomes a security/privacy non-objective.

This whole expression is qualified with a generic level. The generic level is associated with the abstraction/richness of the attributes associated with that security objective. The richness of the attributes increases as we go down from product-> architecture-> design, as explained in Figure 1.

![Security objective root elements](image)

*Figure 1: Security objective root elements*
Secure Development Lifecycle

Security Development Lifecycle (SDL) is a set of activities and milestones which can drive high-quality security outcomes in product and services development at Intel. SDL is Intel's approach to make security and privacy an integral part of our product definition, design, development and validation. SDL integrates with the Intel corporate product life cycle process in order to help ensure that Intel products meet Intel Security and Privacy requirements.

For the purposes of this document SDL can be subdivided into 4 distinct sections: Architecture and Design, Implementation, Validation and Release. After each stage is complete a review is conducted. Below are the activities this release of Unite have gone through:

Architecture and Design
- Threat Model Development and Review
- Secure Architecture Design and Design Review
- Privacy Architecture Design and Review
- 3rd Party Component Review
- API and Interface Review

Implementation
- Review of Input Validation
- Manual Code Review
- Review of 3rd Party Component Risk
- Static Analysis Scans

Validation
- Security Validation Strategy development and review
- Security Validation Execution

Release
- Review for Debug Access Risk
- Vulnerability Review
- Review of Signed Binaries
- Malware/Virus Scan
- Incident Response Plan
Product Architecture Overview

Intel® Unite™ Solution Architecture Overview

The architecture for the Intel Unite solution version 3.0 is largely unchanged from the previous 1.0 and 2.0 versions, diagrams of which can be found below. In previous architectures, ranger, an IIS based server, is used to orchestrate PIN's between hubs and client devices. When a user enters the PIN, ranger connects the user with the version of Unite using the PIN. All communication is encrypted using SSL and uses self-signed certificates generated at launch time by Unite. At launch time Hubs send a hash of their certificate to ranger. When a Hub's PIN is typed in, clients receive the corresponding hash from ranger and verify before making direct connections with hubs. Diagrams detailing this architecture and the flows follow below:

![Unite 1.0 – 2.0 Architecture – IIS based](image)

*Figure 2: Enterprise Unite Architecture*
For the Intel Unite solution version 3.0 the added features are the ability to lock meetings, schedule meetings and audio and video via WebRTC.

**Meeting Lock**

After a meeting is locked, no one is allowed to join, at the end of the meeting when the participants leave the meeting the hub leaves the meeting. So the meeting is no longer valid and is thus not locked. Meeting participants can unlock the meeting while it is still ongoing, returning the meeting/hub to a regular meeting/hub session, where new users can join freely. All users are allowed to lock/unlock the meeting/session so there are not any roles and/or permissions associated with users.
Scheduled Meeting
Scheduled meetings are a way for people to be associated while connecting to a hub. Users will be able to generate/create a meeting that will produce a URL that will launch the application and try to connect to the meeting. The URL can be created via an outlook plugin that will schedule the meeting in Outlook and generate a URL or a website from which users can access and obtain a URL. The meetingID, a globally unique identifier, is what ties URL’s to meetings. The Hub removes itself from the meeting when participant list is zero. The Hub removes itself from a meeting in the case it is still connected on startup as well.

WebRTC support for Video with Audio Capabilities
WebRTC will replace prior mechanisms for sharing screen and will also add video with audio capabilities. After a connection is established between two clients via SSL:

1. Client A sends a command to client B indicating it wants to stream
2. Clients A and B conduct standard WebRTC flow which begins with an SDP offer and answer, handled by managed code used in the application
3. Afterwards direct WebRTC flows are handled by dll’s wrapping open source WebRTC implementation

Intel® Unite™ solution Protected Guest Access Architecture Overview
The Protected Guest Access architecture has been designed to allow guest users to connect and collaborate within the Intel Unite application. The guest does not have access to the customer’s guest network and thus cannot use the ranger server. The Protected Guest Access plugin allows a customer to use the Intel Unite application with a customer’s Hub by turning on the SoftAP and allowing the guest to connect.

The corporate user connects to the Hub and opens the guest access plugin and starts the service. The service starts the Windows Hosted Network (SoftAP) with a default random PSK and new SSID. When the guest connects to the SoftAP he or she can download a special version of the Intel Unite application which comes with a certificate hash used for trust verification between the Hub and the guest. The privileged service verifies signatures on the plugin turning it on to prevent privilege escalation.

The following diagrams detail the architectures and flows described above:
NOTE: No computer system can be absolutely secure. Intel makes no warranty with respect to any malfunctions or other errors in its hardware products or software products caused by virus, infection, worm, or similar malicious code not developed or introduced by Intel. Intel makes no warranty that any hardware products or software products will protect against all possible security threats, including intentional misconduct by third parties. Intel is not liable for any downtime or service interruption, for any lost or stolen data or systems, or for any other damages arising out of or relating to any such actions or intrusions.

Figure 4: Protected Guest Access Architecture
NOTE: The default state is that Secure Guest Access is OFF.
- A Corporate user must be connected to Start.
- When participant list drops to 0, GA is automatically Stopped.
- Guest Access usage is logged and monitored in Admin Web Portal.

Step 1: Corporate User Connects To Hub

START: Corp User opens plugin and "Starts" GA
- Hub Software calls "Start" method on Plugin
- Plugin opens socket on Local host to Guest Access Service and provides random SSID, PSK, and Start command.
- Guest Access Service Starts Windows Hosted Network (SoftAP)

STOP: Corp User opens Plugin and "Stops" GA
- Hub Software calls "Stop" method on Plugin
- Plugin opens socket on Localhost to Guest Access Services and issues "STOP" command
- Guest Access Services Stops Windows Hosted Network (SoftAP)

Figure 5: Protected Guest Access Start and Stop flow
Figure 6: Protected Guest Access guest connection initiation.
## Security Objectives

### Product Security Objectives

<table>
<thead>
<tr>
<th>Threat #</th>
<th>Asset</th>
<th>Actor</th>
<th>Attack point</th>
<th>Technique</th>
<th>Protection required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data in Communication between clients.</td>
<td>Network Adversary</td>
<td>On Machine, On the network</td>
<td>MITM, Snooping</td>
<td>Confidentiality, Integrity, Availability, Authentication</td>
</tr>
</tbody>
</table>

**Mitigation strategy:** Direct communication between clients. Certificate verification before connection initiation. Use of SSL and WebRTC based encryption to prevent snooping.


**Security validation strategy:** Examine communication between clients. Attempt to establish MITM between clients.

**Executed Validation:** Capture and sniff incoming packets on client to verify encryption. Attempt to connect Unite client with client which does not match cert hash given to it by ranger. Attempt to replay valid previously used certificate hash to client. Code review of certificate validation.

**Security validation tool(s):** Wireshark, Nmap, Klocwork, WinPcap, Checkmarx, BetterCap

### Protect data in communication client to service

<table>
<thead>
<tr>
<th>Threat #</th>
<th>Asset</th>
<th>Actor</th>
<th>Attack point</th>
<th>Technique</th>
<th>Protection required</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Data in communication client to service.</td>
<td>Network Adversary, Software Adversary</td>
<td>Client, on the network</td>
<td>MITM, Snooping</td>
<td>Confidentiality, Integrity, Availability</td>
</tr>
</tbody>
</table>

**NOTE:** No computer system can be absolutely secure. Intel makes no warranty with respect to any malfunctions or other errors in its hardware products or software products caused by virus, infection, worm, or similar malicious code not developed or introduced by Intel. Intel makes no warranty that any hardware products or software products will protect against all possible security threats, including intentional misconduct by third parties. Intel is not liable for any downtime or service interruption, for any lost or stolen data or systems, or for any other damages arising out of or relating to any such actions or intrusions.
Mitigation strategy: Certificate pinning on the client or use of DNS SRV record in order to ensure client only communicates with the server. Encrypted communication via SSL.

Assurance practice(s): Security-Based Architecture design, Manual Code Review, Static Analysis Scan

Security validation strategy: Attempt to establish MITM between client and server. Examine communication between client and server to ensure it is encrypted. Ensure certificate pinning mechanisms are sufficient on Windows, Mac and iOS.

Validation: Capture and sniff incoming packets on client to verify encryption. Attempt to compromise certificate pinning mechanism. Detailed code review of DNS SRV record reading. Fuzzing of Mac SOAP record reading.

Security validation tool(s): Wireshark, Nmap, Klocwork, Checkmarx

<table>
<thead>
<tr>
<th>Threat #</th>
<th>Asset</th>
<th>Actor</th>
<th>Attack point</th>
<th>Technique</th>
<th>Protection required</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Data at rest in the database</td>
<td>Network Adversary, Software Adversary</td>
<td>On server</td>
<td>SQL Injection, XML Attacks</td>
<td>Confidentiality, Integrity</td>
</tr>
</tbody>
</table>

Mitigation strategy: Ensure that attackers cannot use injection attacks to exfiltrate information from database. Provide guide to IT deploying Unite so that solution is easy to setup. Entity framework handling of database, .net soap handling.


Security validation strategy: Penetration testing on web admin portal which provides access to the database. Testing of SOAP endpoints. Attempts of SQL injection to exfiltrate information from the database.


Security validation tool(s): Wireshark, Nmap, Burp Suite, Klocwork, Checkmarx
Intel Confidential

NOTE: No computer system can be absolutely secure. Intel makes no warranty with respect to any malfunctions or other errors in its hardware products or software products caused by virus, infection, worm, or similar malicious code not developed or introduced by Intel. Intel makes no warranty that any hardware products or software products will protect against all possible security threats, including intentional misconduct by third parties. Intel is not liable for any downtime or service interruption, for any lost or stolen data or systems, or for any other damages arising out of or relating to any such actions or intrusions.

Prevent unauthorized user from connection to meeting or system

<table>
<thead>
<tr>
<th>Threat #</th>
<th>Asset</th>
<th>Actor</th>
<th>Attack point</th>
<th>Technique</th>
<th>Protection required</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Meeting Id, PIN</td>
<td>Network Adversary</td>
<td>On Machine, On the network</td>
<td>Data exfiltration</td>
<td>Confidentiality, Authentication</td>
</tr>
</tbody>
</table>

Mitigation strategy: PIN's are refreshed every 5 minutes. Exponential back off algorithm on PIN requests and checks. Users cannot join meetings until a client with PIN has added a hub. Meetings can be locked and clients can view other clients within meetings.


Security validation strategy: Attempt to brute force PIN's from ranger. Attempt at guessing PIN's. Meeting lock and client view are part of product validation strategy and do not require security testing.

Executed Validation: PIN brute force.

Security validation tool(s): Wireshark, Nmap, Klocwork, Checkmarx

Guest Access: Protect Data in communication between guest client to Hub

<table>
<thead>
<tr>
<th>Threat #</th>
<th>Asset</th>
<th>Actor</th>
<th>Attack point</th>
<th>Technique</th>
<th>Protection required</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Data in communication between guest client and Hub</td>
<td>Network Adversary</td>
<td>SoftAP</td>
<td>MITM</td>
<td>Confidentiality, Integrity, Authentication</td>
</tr>
</tbody>
</table>

Mitigation strategy: Randomly chosen SSID and PSK for SoftAP. Use of certificate hash for authentication.

Security validation strategy: Verify that the connection between the clients is encrypted and verify authentication mechanism on the guest client.

Executed Validation: Capture and sniff communication between Hub and guest to ensure encryption. Give guest client an incorrect hash and verify checks succeed.

Security validation tool(s): Wireshark, Nmap, Klocwork, Checkmarx, Bettercap

### Guest Access: Prevent unauthorized .exe from being delivered from box to guest

<table>
<thead>
<tr>
<th>Threat #</th>
<th>Asset</th>
<th>Actor</th>
<th>Attack point</th>
<th>Technique</th>
<th>Protection required</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Unite.exe on machine of guest access.</td>
<td>Unprivileged software adversary, Network adversary</td>
<td>SoftAP, Guest Access Machine</td>
<td>MITM, Modification</td>
<td>Integrity, Modification</td>
</tr>
</tbody>
</table>

Mitigation strategy: Prevent attackers from modifying Intel Unite exe on the machine and on the wire to the box. Proper ACL's on the machine. MITM protection given by threat 5.


Security validation strategy: Verify that Unite exe delivered to box cannot be modified by unprivileged software adversary.

Executed Validation: Test and verify ACL's of Unite exe on install.

Security validation tool(s): Wireshark, Nmap, Klocwork, Checkmarx, Powershell
## Architecture Security Objectives

### Protect configuration information

<table>
<thead>
<tr>
<th>Threat #</th>
<th>Asset</th>
<th>Actor</th>
<th>Attack point</th>
<th>Technique</th>
<th>Protection required</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Configuration information at rest on client and DNS service record</td>
<td>Unprivileged Software Adversary, Network Adversary</td>
<td>On Machine, On Ranger</td>
<td>Modification</td>
<td>Integrity</td>
</tr>
</tbody>
</table>

Mitigation strategy: Prevent attackers from compromising the configuration information via modification. Storage of configuration information at rest on client via proper ACL. Proper use of DNS service record to point client at proper server.


Security validation strategy: Attempt to modify XML file without proper rights. Examine and test DNS SRV record.


Security validation tool(s): Custom scripts to modify install files
Addendum

Changes to the system environment
This document relates exclusively to the details of the product or project specified above. This section is designed to provide requested details on how the product in question interacts with the system environment in question.

This product has software components that are installed in standard user directories. Any exceptions to this are listed below:

This software component also adds or makes modifications to the following system attributes and configurations (such as registry entries, firewall settings, digital certificates, kernel mode drivers, and browser plugins):

Registry Keys:
HKLM\Software\intel\Intel® Unite™
HKLM\Software\Wow6432Node\intel\Intel® Unite™
HKCU\Software\intel\Unite
HKCU\Software\Intel\unite\GuestAccess

Services:
Local System account Intel® Unite™ Guest Access Service

The following are uses of cryptography:
Hashing Algorithms: SHA256
Public-Key Algorithms: RSA-2048
SSL Schemes: TLS 1.2 or TLS 1.1

The following is a list of all known third party open source components used in this product:
- BouncyCastle
- .NET Library Components
- WebRTC
Upon uninstallation of this software component, the following files, settings or changes are not removed or reverted to their pre-installation state:

- None