Jason Li

Professor McNair

EE 584-WS1

25 November 2023

Universal Manager: Seamless Management of Enterprise Mobile and Non-mobile Devices

Organizations utilize Mobile Device Management (MDM) to control and secure devices within wireless networks based on security standards. MDM solutions track which devices are connected to the network and monitor their status, location, and configuration modifications. MDM can also remotely configure mobile devices such as through Wi-Fi profiles and VPN settings to make sure they are up to date with the latest standards. Remote access to the devices offered by MDM can also result in better troubleshooting. Security policies being enforced may include password requirements, end-to-end encryption, and two-factor authentication. MDM interfaces can also delete sensitive data should the mobile device be in the wrong hands. Installed mobile apps can be checked to ensure they have corrected app permissions leading to data security. Not only are costs saved and compliance amplified with MDM solutions, but it also makes employees more productive from the management and remote support features. However, complexity is a key factor against MDM, especially with different devices and operating systems. User privacy is also of concern due to remote monitoring and control. MDM utilizes Over-the-Air (OTA) Provisioning to deliver these services such as configurations and software patches remotely and securely. Network parameters and email accounts can be provisioned to a specific geographic location eliminating any manual setup. Device profiles (even to not obvious devices like the Internet of Things [IoT]) and security certificates for authentication can also be streamlined and pushed to employee devices leading to significant reductions in manual operation. Batch processes can also be pushed to the stream of devices lending a hand to automation efforts that push configuration updates. OTA allows for automation and scaling of batch processes and ensures consistency in configuration and security policy pushes. However, once again, user privacy may go out the window and can be of concern if a perpetrator can remote into an employee’s device, which may cause concern to employees. Also, this becomes challenging as a system complexifies and grows as many devices as possible of different operating systems must be supported.

Implementing MDM with enterprise hosts can be challenging due to the vast array of management protocols and architecture. To remedy this, researchers from Motorola Labs and the University of Illinois, Chicago implemented a Universal Manager that integrates a multi-protocol gateway with a Simple Network Management Protocol (SNMP) based enterprise manager to connect intermittent mobile devices with enterprise hosts seamlessly. Mobile devices used in workplaces were owned by companies in the past and were managed by telecommunication operators and not by enterprise IT staff. Any mobile devices were either not connected to the enterprise network or managed limitedly by the Internet Service Provider (ISP) as they were not envisioned to be integrated into the enterprise IT tech stack. Increasing complexity towards device functionality and support for more productive applications led to interest from enterprises to control mobile devices. If a worker can utilize this innovative technology to become more productive, enterprises would be willing to support it. Mobile devices have constraints such as different operating systems, lower resource usage, and lower energy consumption compared to traditional PCs. If a mobile device is integrated into an enterprise network, it would need to be held to the highest standards like any other device such as being dependable, needing zero downtime, detecting faults rapidly with correction, and performing to Service Level Agreement (SLA) standards.

Mobile devices also must be just as secure as wired devices which are complicated by their mobile nature. Handover between several types of networks allows for mobile and traditional devices to exist in the same sphere in which management systems supervise Operations, Administration, Maintenance, and Provisioning (OMAP) requirements, consolidated Authorization, Authentication, and Accounting (AAA) functions, and consolidated OMAP duties. SNMP is not optimal for mobile devices due to its complexity, size, and protocols underneath that need support [2]. Alternate approaches emphasize the resource-light nature of smartphones. Any solutions with multiple management paradigms for each enterprise device and management station can be costly and complex and require good reasoning. There is no overall system integration which makes administration a nightmare, such as if applications are compatible with different operating systems. An SNMP-based unified management platform utilizes a multi-protocol gateway to translate between enterprise management operations from a management station and specific wireless management protocols.

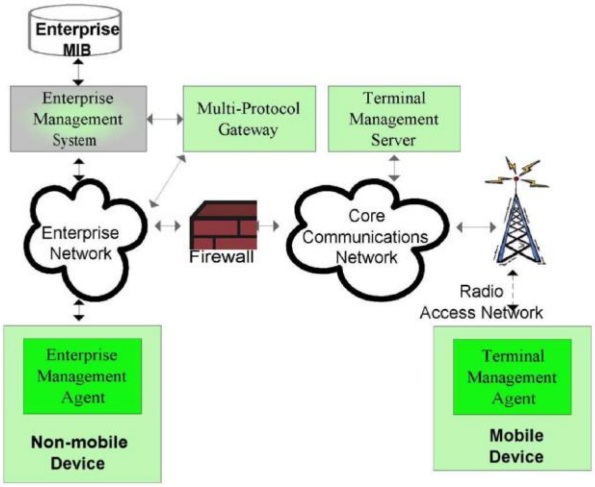

Enterprise MIB
Enterprise Management System
Multi-Protocol Gateway
Terminal Management Server
Enterprise Network
Firewall
Core Communications Network
Radio Access Network
Enterprise Management Agent
Non-mobile Device
Terminal Management Agent
Mobile Device

Figure 1: The Universal Manager Architecture

Figure 1 showcases the Universal Manager Architecture with the enterprise network (user-controlled Personal Area Network (PAN) being connected to a core communications network such as the Internet. A Management Information Base (MIB) stores data regarding the type of device with the Enterprise Management System (EMS) offering devices, hosts, and routers in the enterprise [3]. Discovery mechanisms are used to locate wired enterprise devices and the enterprise management agent pushes any updates to said devices [4]. Mobile devices have a multi-protocol gateway proxy that connects to communicate with the broader network. They also have a Terminal Management (TM) agent that integrates firmware updates such as configuration data, software image installation, and fault and security agents as well as interacting with a TM server with the appropriate protocol. The multi-protocol gateway provides a route for management action to a specific device via mapping IP addresses to subnets to a device's International Mobile Equipment Identity (IMEI) and converting data into proper Protocol Data Units (PDUs) that the device can understand. Through this, the atomicity requirement for this architecture is fulfilled as device updates will not be stalled.

Management protocols have a data model, a data modeling language, data representation, protocol operations, and protocol transports. Since each protocol focuses on a specific subset of features, they have been tailored to specific platforms. SNMP is the IETF standard for Internet operations and maintenance with a manager/agent model having a manager, an agent, a management information database, managed objects, and the network protocol. MIB is utilized with limited commands (e.g., GET, SET) to exchange PDUs. The MIB has a tree structure with individual variables as leaves on the branches. The SNMP data model has an Object Identifier (OID) that makes each MIB and SNMP message unique. SNMP's data modeling language is branched from Abstract Syntax Notation Number One (ASN.1), i.e., SMIv2, and utilizes BER encoding with full implementation in Transport Control Protocol (TCP) or User Datagram Protocol (UDP). Web-Based Enterprise Management (WBEM) is a Distributed Management Task Force (DMTF) standard bundle that models data via a special case of Common Information Model (CIM) through HTTP. CIM has object-oriented modeling with classes and associations akin to Universal Modeling Language (UML). Microsoft System Management Server is built on WBEM and can manage all the Windows systems in a network. XML Based Protocols include BEEP, SOAP, JXTA, and NetConf. BEEP constructs application protocols via XML and MIME (Multipurpose Internet Mail Extension) and can import MIME content. SOAP is an application messaging framework that applies to XML applications by using an envelope structure to organize XML data and an encoding protocol binding framework with support for importing and exporting XML content. JXTA is a framework that overlays peer-to-peer applications with protocols for Peer discovery, binding, and resolving information as well as endpoint routing and identity management. NetConf is a standard that manages network devices currently being used in JUNOScript that executes Remote Procedure Calls (RPC) via SSH, SSL, or telnet through XML representation. Wireless Application Protocol (WAP) Provisioning has the architecture necessary to provision via OTA or smart cards with it first sending a request for acknowledgment which then allows access to remote servers, WAP proxies, or application servers for configuration. SyncML is an XML-based information synchronization standard for wireless devices, making it resource-light as well as language and protocol-neutral by not including features such as server sockets that are inefficient for mobile devices. OEM-specific extensibility is supported as well as implementations of transport protocols like HTTP, WSP, and OBEX. Its popularity led to the extension of SyncML-DM which can manage devices by pushing management actions to management objects which can be device configurations or run-time software environments. Device configurations can have reading and setting parameter keys and values while the run-time software environment consists of software maintenance. Get and Set are completely specific methods that can be complex, i.e., one may need to put in an additional module to get these commands to work. Figure 2 showcases a setup phase to authenticate the mobile device and the management server then a management phase that allows for multiple sessions where information can flow from server to mobile device. A management session can start with Packet 0 if a device is out-of-band.

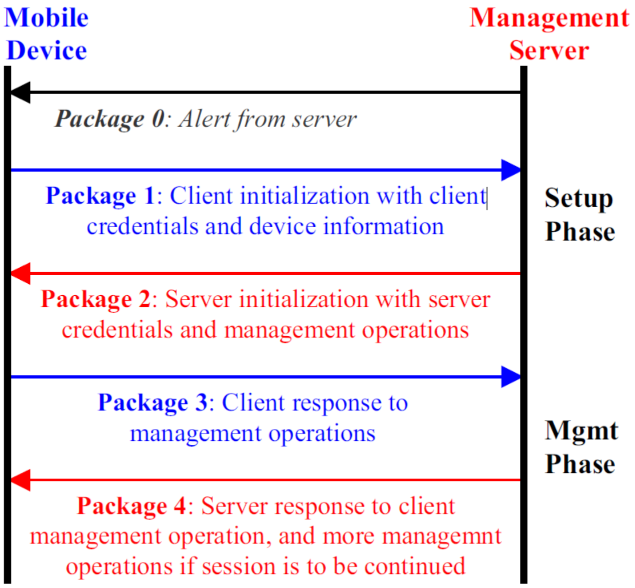
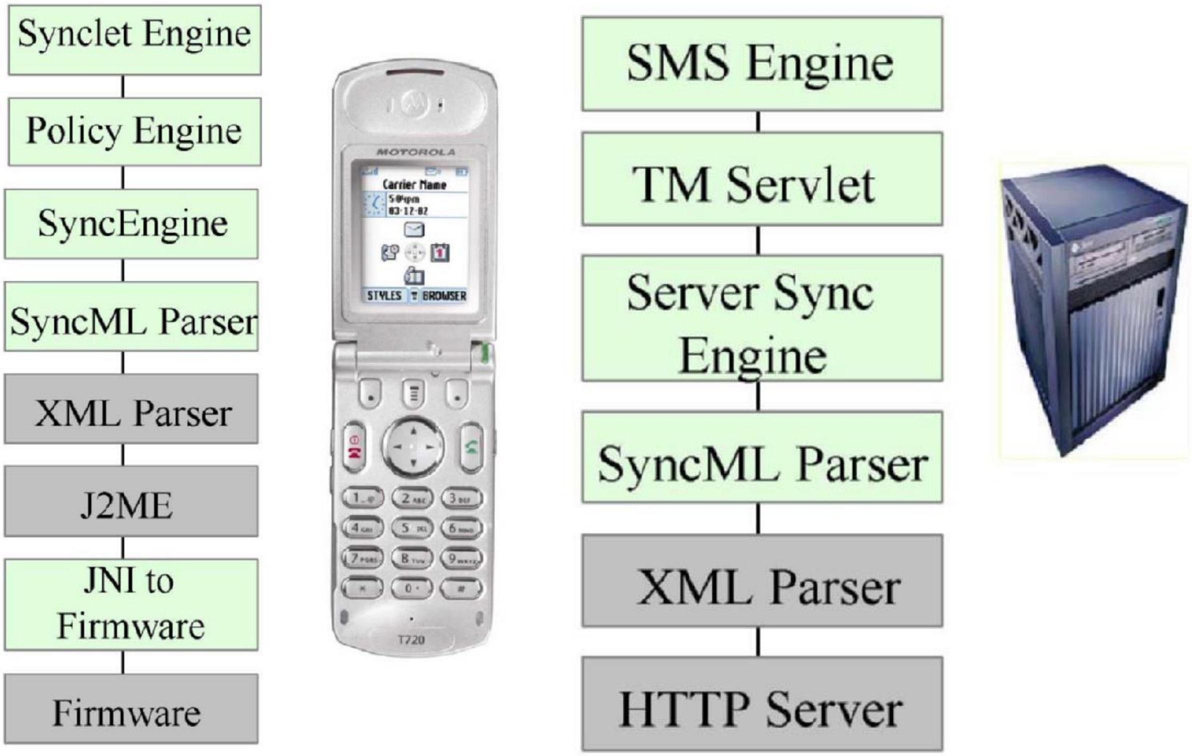

Mobile Device
Management Server
Package 0: Alert from server
Package 1: Client initialization with client credentials and device information
Setup Phase
Package 2: Server initialization with server credentials and management operations
Package 3: Client response to management operations
Mgmt Phase
Package 4: Server response to client management operation, and more managemnt operations if session is to be continued

Figure 2: SyncML Protocol Phases

A multi-protocol gateway is a software entity that is a terminal with its protocol and tricks the enterprise system into accepting its communication stream. The gateway is the intermediary between mobile devices and the enterprise network. Features of the gateway include a protocol converter, mapper, job scheduler, and notifier. The protocol converter converts the enterprise network protocol into the protocol accepted by the wireless device. The mapper has all the mobile devices from discovery mechanism or handshake results mapped to a specific enterprise address space by translating for example a private IP address to a mobile device's IMEI. The job scheduler load balances all terminal activities and optimally schedules TM operations while the notifier is responsible for notifying the enterprise system after a terminal management operation is complete. For most routers and hosts, SNMP is compatible with the enterprise network by having an SNMP agent transfer PDUs between the manager and the device. The Enterprise Management Server also has database capabilities for all the devices via MIBS for each type of device and can display GET operations on specific Object IDs. Wireless terminals do not have the resources to have an SNMP agent which is where SyncML fits in. The multi-protocol gateway can convert SNMP PDUs to SyncML packages including security credentials for the terminal management server. The agent is adapted from the SYMPLE (SyncML Programming Language) platform which adds additional SyncML device management functionality via scripting capabilities and extends the SyncML runtime framework to include scheduling, evaluation, and lifecycle of Symple scripts. Synclets are executable scripts of Symple commands while Synclet specifications have policies and action routines with the former details non-function aspects of a Synclet such as its execution method while action routines are the actual bread and butter that perform the actual intended duty. This structure allows for endless modularization and refactoring. The SyncML parser is an extension of the XML parser that documents the management actions that still need to be executed. Close classes in Java Native were called via methods to train a socket to receive UDP or SMS messages. The Terminal Management server is an HTTP Web server with a Servlet Engine that receives SyncML messages over HTTP and interprets them to actions that are performed. SMS Engine sends SMTP messages via a Java mail plugin for TM Midlets. Otherwise, a UDP converter is utilized to interpret those messages.



Figures 3 and 4: Terminal Management Agent (left) and Server (right).

Terminal management actions are not recognized by the enterprise management system with it either sends data to mobile devices via IMEI and the multi-protocol gateway instinctively or the multi-protocol gateway advertises addressability to IMEIs via mapped IP addresses with terminal management action in a proxy IP address. Both ways have the SNMP PDU sent to the Multi-Protocol Gateway rather than direct transmission with the host and the port of a SNMP is based on the Multi-Protocol Gateway and not that of the device. The SNMP Packet has the Operation Name, the queried OID, optionally information regarding a wireless device, SNMP credentials, and the SNMP Packet ID request. The Multi-Protocol Gateway extracts the SNMP PDUs and checks credentials then maps SNMP OID to a corresponding SyncML management tree node. No IMEI has the destination IP address mapped to the device IMEI. After the Protocol Converter converts the mobile protocol to SyncML, it passes it to the Scheduler to schedule the task on the TM Servlet. HTTP Get is utilized to send the request to the TM Servlet with the Operation Name, SyncML management tree node, SyncML value if it is a Set Operation, the device's IMEI, and the Request ID parameters. The Notifier in the Multi-Protcol Gateway replies with a status to the enterprise management server stating the task is added to the TM SyncML Server to prevent time outs and SNMP PDU retransmissions in port 15001, where the server listens to the Multi-Protocol Gateway. SyncML Servlet oversees all requests from the Gateway and can trigger a mobile device to activate the TM agent and connect to the TM server including credential and capabilities exchange then a SyncML package sends a management action to the agent. Then, the requested operation is executed and sends back a status update to the TM server. The Notifier also has a Poller component (as the Enterprise Management System is behind a firewall) that polls the TM Server via HTTP Get to see if values are got from the mobile terminal and then queries the TM server for results. If received from the phone, the TM Server sends them via HTTP to the Multi-Procotol Gateway. The Protocol Converter converts TM Server results to SNMP PDUs and sends them to port 150002 which has the enterprise management server listening with the request Command (Get/Get, Next/Set), request ID with the SNMP Packet, SNMP OID, value connected to OID, and delivery status. This information is extracted and added to the actual application.

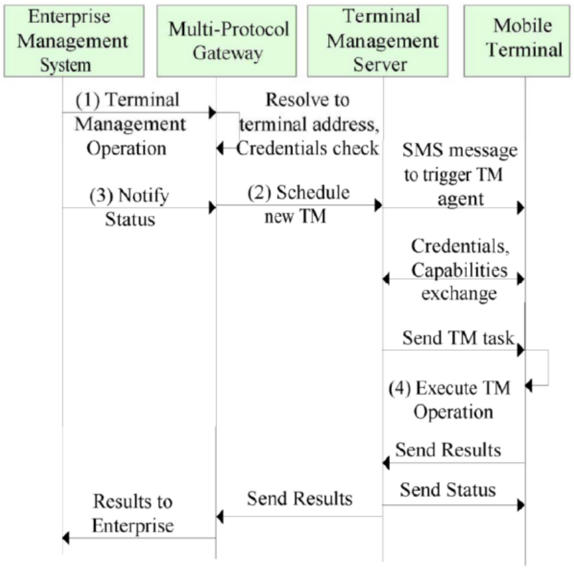

Enterprise Management System
Multi-Protocol Gateway
Terminal Management Server
Mobile Terminal
(1) Terminal Management Operation
Resolve to terminal address, Credentials check
SMS message to trigger TM agent
(3) Notify Status
(2) Schedule new TM
Credentials, Capabilities exchange
Send TM task
(4) Execute TM Operation
Send Results
Send Status
Results to Enterprise
Send Results

Figure 5: Flow of operations

The security assessment involves identifying the assets, perpetrators, threats, safeguards, vulnerabilities, and additional controls. Assets can include resources, storage, and identification information of everybody in the network and the Enterprise Management System as this can be used by perpetrators to continue blackmail through phishing overs or through social engineering, look for valuable data that can be sold or used as a front to compromise as a dummy account, or to identify more about how the network and system operates. These reasonings can also apply to data found on each mobile device including specific identifiers like IMEI. The agent and communication method can also be breached and exploited for the bandwidth to try to crack any data or to gain recognizance intelligence regarding how the signal plays into the larger system. Perpetrators can include competing companies to find out secrets regarding the system's technology that can be used to reverse engineer their solution or even damage the reputation of the company being attacked. Cybercriminals, either by themselves or backed by a crime syndicate or foreign government, also commonly attack enterprise networks to either mine data or cause a system shutdown. The former is to take any company resources that can be useful for them or to be resold and to gather data that they can use later to continue their nefarious tasks while the latter is akin to causing chaos amongst the users of the system, whether they are internal employees or external clients like stockholders. A disgruntled employee with domain and firsthand knowledge of the enterprise network system can also be a risk especially if they are still working on the system. To get back at any injustices, whether personal or companywide, they can do internal damage that may be tough to fix if the employee is well versed in the system. Threats include impersonating a mobile device to gain access to the network which could be possible. Another threat includes brute forcing into the enterprise network by bypassing the firewall setup or via having an internal connection to verify and authenticate a perpetrator's identity. The protocols being used whether data transmission protocols like SMS or TCP/IP or the management action protocols such as SyncML or SYMPLE can be prone to weaknesses. Packets could be tampered with to forge data and allow for entry for an unauthorized mobile device. Management action protocols can also have their commands modified to perform operations that could be lethal. Safeguards involve a firewall that blocks traffic except for two ports that can communicate to mobile devices and a Security Information and Event Management (SIEM) that can integrate with the Enterprise Management System and track suspicious activity of connected mobile devices. These safeguards should limit and monitor a good chunk of attacks but not all of them. Vulnerabilities include the process of registering a mobile device as a perpetrator could fake authentication to breach the network. Whether it is through brute forcing configuration settings or exploiting embedded system logic in the Terminal Management Server, these are the ways a mobile device can be spoofed. In the wireless communication realm, all signal processing exploits can be employed by perpetrators including shadowing, fading, jamming, or tuning into the frequency. The perpetrator aims to deny service in the case of jamming to try to get a good signal to either snoop or gain access to the system. If a back door is found and breached, the network can easily be shut down with the lack of an intrusion detection system or load balancers in the network. Additional controls involve including Advanced Encryption Standard (AES) with either a 128 or 256 initialization key, implementing load balancers and intrusion detection systems, as well as making architectural revisions to emphasize reliability, availability, and robustness in the system.

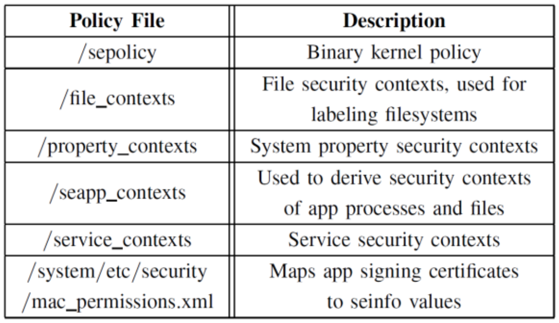
The major security issues range from vulnerabilities in the authentication process, to stress on the enterprise network at hand, and the lack of protection from a firewall. As mentioned before, the authentication process is executed via handshake, but the paper does not detail whether the handshake protocol is robust. Tampering with the packets that are transmitted between the mobile device and the TM server with the agent can result in unauthorized access to the system at large. A perpetrator can also inject malicious code into the agent or any of its other protocols via a weak backdoor to get the registration process to allow them in. This also works with taking advantage of any programming or logic oversights such as a buffer overflow. The paper also does not stress the risk of whether a network gets flooded with too much activity as a perpetrator can institute a Distributed Denial of Service (DDoS) attack by registering many devices under pretenses to shut down the network. Lastly, the firewall exists in the enterprise network yet there are no other security measures put into place. This is concerning as the Poller, an important part of the Universal Manager, is located behind the firewall. Without the firewall at hand, notifications regarding device registration can be in serious jeopardy. In addition, one firewall and no other verification points at other areas of the network make the enterprise vulnerable to snooping beyond what a perpetrator initially gained access to.

The security concerns are dramatic, especially with no safeguards in place. Without proper monitoring of the network via the Enterprise Management System, many foul plays can be enacted to cause damage or steal valuable resources from the enterprise network. The methods a perpetrator will use are brute forcing their way into a valid input to get in as well as performing outside surveillance to see how the whole enterprise network is configured. Doing so are methods that take the least amount of resources and/or can be done with a bunch of outside research and reverse engineering. The methods are likely to involve finding backdoor vulnerabilities or breaching by abusing the registration process. This takes more time and resources compared to the more likely options and it takes wit, intelligence, and a lot more strategy to gain access. The least likely options of attack are those that take way more resources than the other strategies which include signal tampering, DDoS, or injecting malicious code into the entire system. This takes the most wits and resources in execution as signal tampering involves having high-quality equipment to achieve the function, DDoS requires many mobile devices, and malicious codes require understanding where in the actual software programming what is vulnerable.

Security can be improved in all aspects. On an overall architecture level, the enterprise network can be redesigned to emphasize availability, robustness, and reliability. No one server should exist containing all the data as there should be at least one backup accessible should a vital node go down. In addition, connections can be expanded so there is not only one route that data must go through to reach a certain node. Checkpoints should also be implemented around the network to verify that access to nodes, especially critical ones, is valid. Authorization can be immensely improved through two practices: the principle of least privilege and encrypting all communications. Big picture-wise, identity, and access management should be implemented that can be integrated into the Enterprise Management System with each mobile device registered should be tied to the identity of someone in the enterprise. No one person should have access or permission to domains that they do not need to interface with; doing so prevents any insiders from tampering with any critical systems. Encryption within the network as well as through the mobile agent should have the AES-128 or AES-256 to guarantee the best protection against any cryptographic attacks. Intrusion detection and network monitoring tools can also be integrated with the Enterprise Management System to track all mobile devices in the network and make sure that there is no abnormal activity.

Mobile Device Management has now become more common due to the propensity of an employee's desire to bring their device. For corporations that value security, employees are not allowed to work on enterprise work from their laptops, so they provide laptops to all employees based on specific security needs and requirements. However, enterprises have embraced Mobile Device Management for employees' smartphones primarily to increase productivity in non-confidential items such as email, calendar, and team messaging [5]. Security is built in from good software engineering practices that minimize security loopholes that perpetrators can access through having a secure network. In addition, mobile device registration is started from the enterprise network and not externally, making it not directly accessible to perpetrators. Employers realize the convenience of Mobile Device Management on smartphones and have thus invested a lot in not only offering the service but also securitizing it. With advances in enterprise network security, transmission protocols, and encryption standards, Mobile Device Management can become tampered with less and less vulnerable to internal and external attacks. The concerns are driven by the initial paper's lack of security details on how the mobile device registration process and the Enterprise Management System are securitized. This was only an acute issue as Mobile Device Management has proliferated through all Information Technology (IT) tech stacks, it has become imperative to offer both convenience and security as well [6]. The author of this final report interfaced with MDM in their summer internship as there were vast two-factor authentication to register the device. In addition, any access to the application even if the author switched to another page necessitated a password to access the network to prevent any unauthorized access.

Security models have already been seen being integrated into mobile device management such as the Android Open-Source Project (AOSP) and SELinux. Researchers from two Chinese universities in 2017 developed a mobile device management system branched from AOSP and SELinux that had real-time SELinux security policy updates. Android is developed via application sandboxing where apps are quarantined to their play area with Linux's discretionary access control (DAC) controls. The National Security Agency (NSA) created SELinux and provided global fine-tuned privilege control via the MAC configurations of the Linux kernel with an open-source project SEAndroid denying app vulnerabilities and promotion attacks that SELinux could not do. Each kernel has a security server in its operating process that has security policies to vow or disavow kernel objects at runtime. Through this, pushes, location tracking, APK file verification, and permission control are boosted from SEAndroid [7].



Policy File	Description
/sepolicy	Binary kernel policy
/file_contexts	File security contexts, used for labeling filesystems
/property_contexts	System property security contexts
/seapp_contexts	Used to derive security contexts of app processes and files
/service_contexts	Service security contexts
/system/etc/security /mac_permissions.xml	Maps app signing certificates to seinfo values

Table 1: Android SELinux Policy Files

Researchers from Universiti Malaya in Malaysia have also prepared an Information System Audit scheme for Mobile Device Management. Vulnerabilities are monitored as well as maintaining strict Confidentiality, Authentication, and Integrity policy. The paper also notes how all parties need to understand the risks and vulnerabilities otherwise the controls would be useless. Infrastructure-based protection is essential especially for network, system, and end-device security as well as network segmentation to isolate channels of data. MDM allows blocking domains as well as its monitoring capabilities of security. ISO, NIST, and ITIL standards are in place to support connection security, data access, application, and physical aspects of a device [8]. Perpetrators are always getting smarter and thus MDM providers need to rely on the most promising security research to better securitize their product at hand.

References

[1] Sandeep Adwankar, Sangita Mohan and V. Vasudevan, "Universal Manager: seamless management of enterprise mobile and non-mobile devices," *IEEE International Conference on Mobile Data Management*, 2004. Proceedings. 2004, Berkeley, CA, USA, 2004, pp. 320-331, doi: 10.1109/MDM.2004.1263082.

[2] J. Case, M. Fedor, M. Schoffstall, J. Davin. “A Simple Network Management Protocol.” RFC 1157, *Internet Engineering Task Force*, May 1990.

[3] K. McCloghrie, M. Rose. Management Information Base for Network Management of TCP/IP-based internets:MIBII. RFC 1213, *Internet Engineering Task Force*, March 1991.

[4] W. Stallings. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. 3rd ed. *Addison-Wesley*, 2000.

[5] “Mobile device management.” *Wikipedia*. https://en.wikipedia.org/wiki/Mobile_device_management (accessed November 13, 2023).

[6] Elaine Atwell. “The Pros and Cons of Mobile Device Management (MDM) Solutions. *Kolide*. https://www.kolide.com/blog/the-pros-and-cons-of-mobile-device-management-mdm-solutions (accessed November 13, 2023)

[7] X. -l. Song and C. -H. Yang, "Mobile Device Management System Based on AOSP and SELinux," *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC),* Shenzhen, China, 2017, pp. 417-420, doi: 10.1109/DSC.2017.93.

[8] N. A. Abu Othman, A. A. Norman and M. L. Mat Kiah, "Information System Audit for Mobile Device Security Assessment," *2021 3rd International Cyber Resilience Conference (CRC)*, Langkawi Island, Malaysia, 2021, pp. 1-6, doi: 10.1109/CRC50527.2021.9392468.