Pers Ubiquit Comput (2018) 22:23–34
https://doi.org/10.1007/s00779-017-1098-x

ORIGINAL ARTICLE

Resiliency of mobile OS security for secure personal ubiquitous computing

Seongkee Lee[1] · **Sanghoon Lee**[1] · **Taein Kang**[1] · **Miyoung Kwon**[1] · **Nohbok Lee**[1] · **Hoonkyu Kim**[1]

Received: 27 September 2017 / Accepted: 4 November 2017 / Published online: 9 December 2017

Abstract As computing devices such as smartphone are used widely, people conduct their businesses using devices and even enjoy entertainment anywhere. On the other side, worries about privacy or economic damages by cyber attacks are increasing. Although many cyber threats may happen, it is difficult to detect and defend against them before attacks occur, and also it isn't easy to cope with certain attack by one matching defense technique. One approach to solve these problems is to enhance the security of OS (operating systems). We developed a kernel-level mobile OS security technique, called by DMOS (Deep Mobile OS Security), for secure personal ubiquitous computing. It has deep security ability that blocks attacks layer by layer in a defense-in-depth manner so that important content is protected and essential services can be continued even though attacks intrude into the devices. In order to assess how well DMOS can realize such defense abilities, this paper tries to analyze the resiliency capability of DMOS. Referring to the cyber resiliency framework, we analyze the techniques and the defense effects related to resiliency which DMOS can support along cyber attack cycle. Also, we test the resilient defense ability of DMOS under typical cyber attacks scenarios. From analysis and test results, it can be concluded that DMOS has the resiliency capability to realize deep security for personal ubiquitous computing.

Keywords Secure OS · Mobile security · Cyber resiliency · Cyber attack life cycle

✉ Seongkee Lee
seongkeel@add.re.kr

Sanghoon Lee
shljhl@add.re.kr

Taein Kang
tanekang@add.re.kr

Miyoung Kwon
kmyadd@add.re.kr

Nohbok Lee
nblee@add.re.kr

Hoonkyu Kim
hunk@add.re.kr

[1] The 2nd RD Institute, 3rd Directorate, Agency for Defense Development, Seoul, South Korea

1 Introduction

Today, computing environments are extended and dependency on computing services is deepening. Recently, even homes are being digitalized and automated. Home electronic appliances are connected electronically and controlled automatically using IoT technique, smartphone, etc. We can enjoy entertainment such as games or movies and access diverse information on the Internet through WiFi devices at home. On the other hand, worries about privacy or economic damages by cyber attacks are increasing too. When malicious code is concealed in a game and executed on a laptop connected to the network, private information such as addresses or messages may be leaked. Also, harmful contents may be stored and displayed on these devices. Although these threats may happen, since cyber attacks happen at random and covertly, it is difficult to detect and defend against them before attacks occur. Furthermore, it isn't easy to block certain attack with one matching defense technique.

We conceive that one approach to solve these problems is to enhance the security of OS (operating systems). Since OS must be used in most computing environments and controls computing resources as well as services on them, a secure OS

will make computing environments secure. As one essential technique to enhance OS security, we developed a mobile OS security technique, called by DMOS (Deep Mobile OS Security), for secure personal ubiquitous computing. As personal ubiquitous computing needs are increasing gradually, mobile security becomes more important [1–3]. DMOS is a prototype of kernel-level mobile OS security technique which provides smartphone users with secure computing infra and mission services. In particular, it has deep security ability that blocks attacks layer by layer in a defense-in-depth manner so that important contents are protected and essential services can be continued even though attacks intrude into devices.

The purpose of this paper is to assess how well DMOS can realize such deep defense ability. For systematic analysis, we refer to the cyber resiliency framework proposed by MITRE [4] which guides how to examine cyber resiliency of security systems. From a wide perspective, the cyber resiliency is defined as follows: *The ability of a nation*, *organization*, *or mission or business process to anticipate*, *withstand*, *recover from*, *and evolve to improve capabilities in the face of*, *adverse conditions*, *stresses*, *or attacks on the supporting cyber resources it needs to function*. This resiliency definition may coincide with the deep defense ability property. According to this point, we analyze the resiliency capability of DMOS based on the framework, and then from the results, we try to assess the deep defense ability of DMOS.

The cyber resiliency framework specifies the resiliency techniques, cyber attack life cycle, and the resiliency effects which can occur during the cyber attack life cycle when the techniques are applied. Among the resiliency techniques and the effects specified in the framework, this paper identifies the resiliency techniques which DMOS can support and the resiliency effects which DMOS can provide along the cyber attack cycle. Analyzing the results, we extract the resiliency characteristics of DMOS. In real, in order to verify the resiliency capability of DMOS, we test how DMOS can defend against diverse cyber attacks under serious cyber attack scenarios.

In detail, this paper describes the related works, analysis, and testing as follows. First, as related works, in Section 2, we explain the major techniques of mobile OS security technique DMOS. Section 3 defines the cyber resiliency framework by integrating cyber resiliency engineering process, the cyber attack life cycle, and resiliency effects. Following on the framework, in Section 4, we identify the techniques related to mobile security in the framework and examine which techniques of DMOS can support them. Section 5 analyzes the resiliency effects which can occur when DMOS techniques are used during the cyber attack life cycle. Then, from the results, we analyze the resiliency characteristics of DMOS. Finally, by testing the resiliency capability of DMOS under cyber attack scenarios on a smartphone, we verify the practical defense ability of DMOS.

2 Features of the mobile OS security technique DMOS

The mobile OS security technique DMOS is a kernel-level security technique for mobile phones using the Android platform. Major technical features of DMOS are as follows [5].

– Defense-in-depth mechanism. Android consists of multiple layers such as the user, application, framework, and kernel. DMOS adds defense techniques to each layer, for example, user certification in the user layer, monitoring and control in the application layer, app control in the framework layer, and system resource access control in the kernel layer. By this mechanism, when certain attack penetrates a layer, other layers try to block the attack. As a result, the attack is detected or its power may be degraded. The defense-in-depth mechanism can be an effective defense scheme to enhance security ability [6]. Figure 1 shows the defense-in-depth mechanism of DMOS.

– Role-based access control (RBAC) mechanism. Access control limits access to computing assets and services. DMOS adopts the role-based access control mechanism. This mechanism defines a security policy by subject-role-object type. When a subject (process or app) requests a certain role (read, write, execute, delete, etc.) to an object (process, file, or data), the security policy decides whether to allow or deny the request. Since it is possible to control the permission to operation level, even when rooting is happening, if the operation to target is not allowed by policy, then the attack on the target fails. Figure 2 shows the access control mechanism of DMOS. By this RBAC

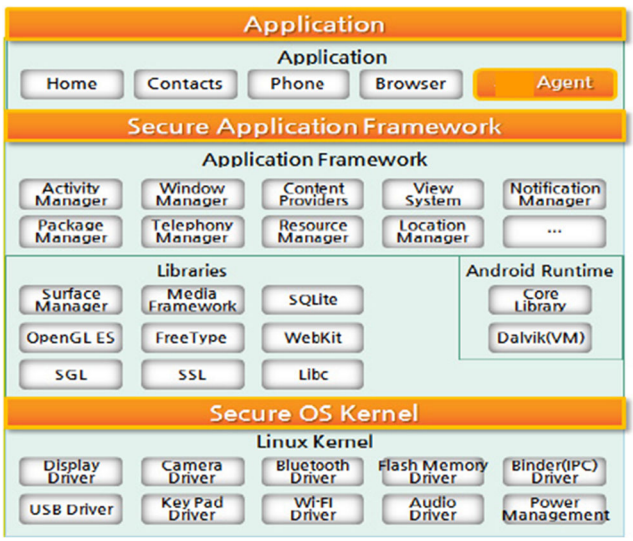


Fig. 1 Defense-in-depth of DMOS

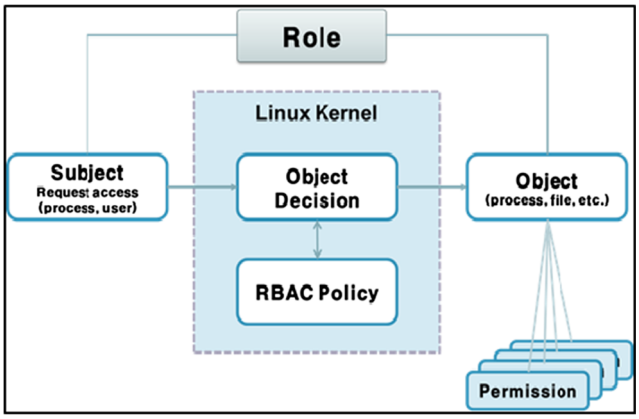


Fig. 2 Role-based access control of DMOS

mechanism, only permitted services can be executed on the mobile phone. Any services not permitted can't be executed on the phone. This mechanism is a strong mechanism that expunges unnecessary stuffs which attacker may use to intrude into the phone.

- Mobile device management (MDM). DMOS certifies both users and mobile devices. Uncertified users can't use any services on the phone, and also uncertified devices also can't be used. MDM monitors all usage status and logs the events related to security such as invalid access behaviors or processes. In special cases, when devices are lost over the specified time durations or the designated locations, they can be locked and data on devices can be wiped out to thoroughly block attacks. Such management techniques are executed on the management server. The server delivers and applies the security policies to mobile devices.
- Other features. DMOS can conceal files and data so that attackers can't find stuffs which may be used to intrude into the device. This delays or deters attacks. Also, sensitive data is encrypted, so even when devices are stolen by an attacker, important data isn't readable. In addition, in order to find whether the kernel or files are forged, the system integrity is checked at runtime. This feature helps detect the indications of an attack in advance.

The technical features described above are implemented into the secure kernel, framework, and management techniques of DMOS like the following.

Secure Kernel (*K*) *techniques*

K1-SEPOL: apply security policy in kernel
K2-FHOOK: control (deny/allow) file creation, read, write, and execution
K3-FCHCT: control file size, name, owner, and privilege change of file
K4-CONCL: conceal directory and process
K5-RESCT: control change of resources
K6-PROCT: control termination, priority, trace of processes
K7-SYSCT: control memory access, system off, file mount, USB connection, and kernel booting
K8-SETID: control SetID execution and privilege escalation
K9-FENDE: encrypt file at the time of file storage
K10-INTEG: check file integrity at execution and at periodic

Secure Framework (*F*) *techniques*

F1-SEPOL: apply security policy in framework
F2-APPCT: control installation, execution, and techniques of apps
F3-HWPER: control usage of camera, GPS, WiFi, NFC, and mike
F4-APPCH: check forgery of app installation/execution files
F5-CERTI: certify mobile devices and users
F6-DEVMT: lock device, wipe data out, and infer stolen/lost device; monitor device status
F7-AKFIF: check transmission of MDM command, events and security policy files btw kernel and framework

Secure Management (*M*) *techniques*

M1-APPMT: register, retrieve, and delete apps
M2-DEVMT: register, retrieve, and delete devices
M3-USRMT: register, retrieve, and delete users
M4-POLED: create, edit, and retrieve security policy
M5-LOGTR: receive, retrieve, and delete log
M6-HWONF: on/off WiFi/GPS/camera/mike/bluetooth
M7-ACTCH: check MDM commands execution
M8-MONDS: display security alarms and MDM information with management messages

These techniques can play a role to enhance the security of mobile computing, a part of personal ubiquitous computing. In order to assess the defense capability of DMOS techniques, as described in Section 1, this paper analyzes how well DMOS techniques can resiliently cope with cyber attacks.

3 Cyber resiliency framework

Although there are many works related to cyber resiliency [7–14], few are related to the systematic works on cyber resiliency assessment. Uniquely, MITRE proposed two

procedures to help assess the cyber resiliency. One procedure defines the cyber resiliency engineering process with resiliency techniques [15], and another procedure specifies the resiliency effects when the resiliency techniques are applied to the cyber attack life cycle [16]. When integrating two procedures, we can establish a systematic cyber resiliency framework.

3.1 Cyber resiliency engineering process

Cyber resiliency engineering proceeds from goals and objectives into techniques. At first, the goals define the ultimate intended resiliency capabilities as follows: anticipate, which maintains a state of informed preparedness; withstand, which continues essential techniques; recover, which restores techniques; and evolve, which adapts techniques into changes. The objectives are more detailed capabilities that achieve goals as follows: understand, prepare, prevent/avoid, continue, constrain, reconstitute, transform, and re-architect. Finally, the technique part specifies the technique groups needed to achieve the objectives. Here, each group includes the detailed techniques related to resiliency. Figure 3 shows the cyber resilient engineering process.

3.2 Cyber attack life cycle and resiliency effects

Cyber attack has a certain life cycle separated into several steps as follows [16, 17]: (1) reconnaissance step, in which attacker identifies target and plan attack activities; (2) weaponize step, in which an attacker develops an exploit and makes it into a deliverable form; (3) deliver step, in which an attacker delivers the exploit to a target system,; (4) exploit step, in which exploits are installed on the initial target; (5) control step, in which an attacker employs mechanisms to control the initial target and compromises additional targets; (6) execute step, in which an attacker executes the attack plan and achieves the attack objectives; (7) maintain step, in which an attacker ensures a sustained presence on the compromised targets and may erase prior attack activities.

When cyber resiliency techniques are applied on the steps of the cyber attack cycle, diverse resiliency effects can happen as follows [16]: detect, deter, delay, degrade, deceive, contain, curtail, negate, recover, etc. Figure 4 shows that cyber resiliency techniques are applied on the attack steps of the attack life cycle, and at that time, the resiliency effects can occur.

In Figs. 3 and 4, cyber resiliency techniques specify the technique groups in which the detailed techniques related to resiliency are included. Table 1 shows the detailed techniques belonging to each group. A total of 44 detailed techniques in 14 technique groups are specified.

Based on the cyber resiliency framework established above, this paper analyzes the cyber resiliency of the DMOS in detail.

4 Cyber resiliency techniques in DMOS

4.1 Identifying resiliency techniques in DMOS

Among cyber resiliency technique groups in Table 1, adaptive response, diversity, and unpredictability technique groups are not included in DMOS technical requirements. Also, the detailed techniques checked by double asterisk may not be applicable to mobile security. Therefore, only the 27 detail resiliency techniques are related to resiliency of mobile security. For the 27 resiliency techniques, we check whether DMOS can support them or not and find out that DMOS can support 20 detailed techniques. That is, with 20 techniques among 27 resiliency techniques, about 74% of cyber resiliency techniques related to mobile security in the framework, can be supported by DMOS.

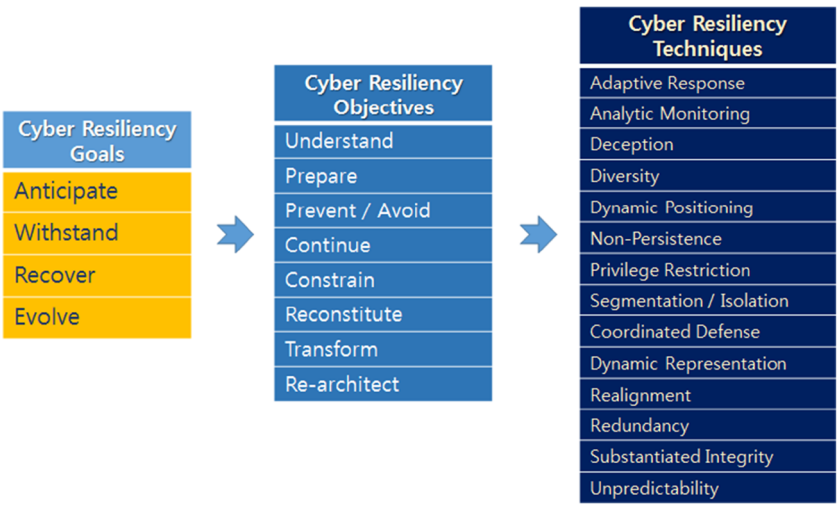


Fig. 3 Cyber resiliency engineering process

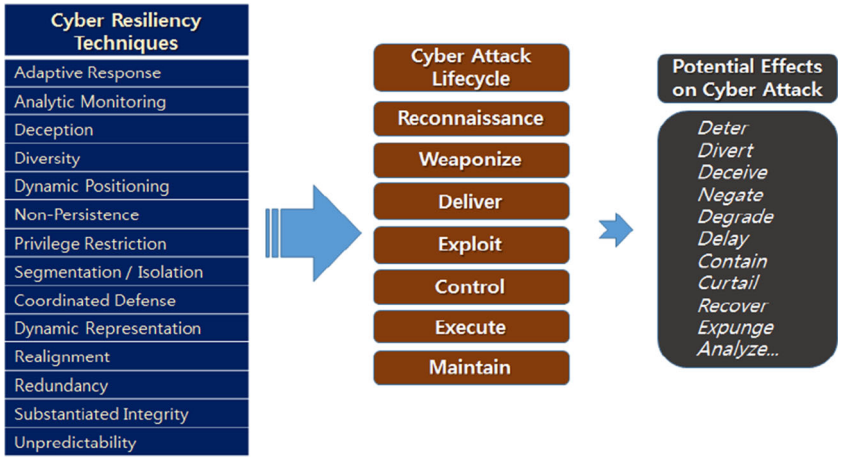


Fig. 4 Cyber attack life cycle and resiliency effects

4.2 Resiliency techniques in DMOS

In order to assess the resiliency capability of DMOS, it is first necessary to check to what extent the 20 resiliency techniques can be realized by DMOS techniques. By examining the secure kernel, framework, and management techniques of DMOS, we analyze which techniques of DMOS can be used to satisfy the 20 resiliency techniques. Table 2 shows the result. Each resiliency technique in the framework can be realized by one more techniques rather than one technique in DMOS. For example, monitoring and damage assessment technique can be realized by F6-DEVMT and M8-MONDS techniques in DMOS, and also obfuscation technique can be realized by K4-CONCL and M5-LOGTR in DMOS. Inversely, one technique of DMOS can be used to support several resiliency techniques.

In order to assess cyber resiliency objectively, it may be effective to calculate cyber resiliency in quantitative measurement methods such as resiliency metric. Bodeau et al. [18] propose some metrics to help measure the resiliency of a technique. However, there are several constraints in applying these metrics practically. One of them is that many assumptions should be solved in order to use the proposed metrics beforehand. For example, in order to calculate the Metric-47 defined in [18], that is, the average length of time between the occurrence and the discovery of an anomaly, it must be possible to define the anomaly state, identify when an anomaly occurs, and when an anomaly is discovered. Actually, it isn't easy to satisfy these assumptions without real operational environments. Another constraint is that even if all assumptions are satisfied, it isn't clear to what extent the measured values imply resiliency level. The criteria to evaluate the resiliency level for each metric must be clearly defined. By these reasons, the metrics are not applicable to measure the cyber resiliency yet. Therefore, as defined in the cyber resiliency framework, it is a more practical approach to analyze which techniques in DMOS can be used on which steps of the attack cycle and which resiliency effects can occur at that time. Section 5 discusses that in detail.

5 Assessment of cyber resiliency of DMOS

5.1 Resiliency effects on cyber attack life cycle

As described in the framework, the types of cyber resiliency effects are the following: analyze effect, which judges attack damages; detect effect, which finds attack activities; deter effect, which cuts attack intention; delay effect, which reduces attack speed; degrade effect, which drops attack power; deceive effect, which confuses the attacker; contain effect, which restricts attack activities; curtail effect, which limits attack effectiveness; negate effect, which prevents attack activities; recover effect, which backs up damage; and expunge effect, which deletes unnecessary services.

The framework pre-specifies the resiliency effects that each resiliency technique can support on the steps of the attack cycle. For example, as shown in Table 3, the monitoring and damage assessment technique can support *detecting* attacks on reconnaissance, the deliver and maintain step help *analyze* attack damage on the execution step. Since the monitoring and damage assessment technique is implemented by F6 and M8 techniques in DMOS as analyzed in Table 2, we can naturally reason that F6 and M8 techniques provide the detect effect on reconnaissance, deliver and maintain steps, and the analyze effect on the execution step. Similarly, K4 and M5 techniques supporting obfuscation can be applied to delay and degrade reconnaissance activity and degrade execution activity of the attacker. Actually, because the target can't be displayed by K4 directory concealment and M5 log manipulation techniques, the attacker may not find the target. This results in the attack being delayed or degraded. Table 3 shows all the resiliency effects which DMOS techniques can provide on the steps of the attack cycle.

Table 1 Resiliency techniques in framework

Technique groups	Detailed techniques
Adaptive response*	Dynamic reconfiguration, dynamic resource allocation, adaptive management
Analytic monitoring	Monitoring and damage assessment, sensor fusion and analysis**, Malware and forensic analysis
Coordinated defense	Technical defense-in-depth, coordination and consistency analysis
Deception	Obfuscation, dissimulation/disinformation**, misdirection/simulation**
Diversity*	Architectural diversity/heterogeneity, design diversity/heterogeneity, synthetic diversity, information diversity, C3 path diversity, supply chain diversity
Dynamic positioning	Functional relocation of sensor**, functional relocation of cyber assets**, asset mobility, distributed functionality
Dynamic representation	Dynamic mapping and profiling, dynamic threat modeling, mission dependency and status visualization
Non-persistence	Non-persistent information, non-persistent services, non-persistent connectivity
Privilege restriction	Privilege management, privilege-based usage restriction, dynamic privileges
Realignment	Purposing, offloading/outsourcing, restriction, replacement
Redundancy	Protected backup and restore, surplus capacity, replication
Segmentation/isolation	Predefined segmentation, dynamic segmentation/isolation
Substantiated integrity	Integrity/quality checks, provenance tracking**, behavior validation
Unpredictability*	Temporal unpredictability, contextual unpredictability
14 techniques group	44 detailed techniques

From Table 3, some notable points about DMOS techniques can be extracted as follows.

- Only the technical defense-in-depth technique can delay the weaponize step. This means that without defense-in-depth technique, there is no way to hinder the weaponize activities of the attacker. K1-SEPOL, K9-FENDE, F1-SEPOL, and F5-CERTI techniques of DMOS are the techniques applicable to cope with the weaponize step of attack.
- The remarkable point is that DMOS provides the techniques to response to all the steps of the attack life cycle from reconnaissance to maintain step. In detail, 15 techniques are provided at the reconnaissance step, 4 at weaponize, 13 at deliver, 16 at exploit, 19 at control, 22 at execution, and 19 at maintain step. This can be interpreted that DMOS has at least the potential capability to defend against cyber attacks resiliently.
- On the whole, the techniques related to the privilege control such as privilege management, privilege-based usage restriction, and restriction play a strong role in blocking attacks, especially after the deliver step.
- After the execute step, many defense techniques are still sustained. This is intended to continuously find cyber threats or prevent unknown cyber attacks even in ordinary time.
- When reorganizing Table 3 in the standpoint of resiliency effect, we can analyze which techniques of DMOS can be supported for each resiliency effect. For example, in order to detect an attack, 7 kernel techniques, 3 framework techniques, and 4 management techniques are applied at the reconnaissance, deliver, control, execute, and maintain step. Also, in order to delay and degrade attacks, about 20 techniques including 9 kernel techniques, 5 framework techniques, and 6 management techniques can be applied on the whole steps of an attack cycle. These analysis results can be also used to check the preparedness of cyber security in certain computing environment. For example, it can check which steps are the most vulnerable and which techniques should be reinforced or newly developed.

5.2 Cyber resiliency characteristics of DMOS

In order to analyze the resiliency characteristics of DMOS, we observe the results above more quantitatively with respect to two aspects: (1) to what extent the techniques of DMOS can response to each step of the attack life cycle and (2) to what extent the techniques of DMOS can contribute to each resiliency effect.

For the first aspect, we examine the secure kernel, framework, and management techniques of DMOS applicable to each attack cycle in detail. Figure 5 shows the result. Secure kernel techniques are used more than other techniques on all steps. In average, 6.4 kernel techniques are applied to block each step of the attack. This is about 2 higher in comparison with other techniques. Also, many management techniques are applied at the reconnaissance step and after the control step. This is to help defenders

Table 2 Resiliency techniques in DMOS

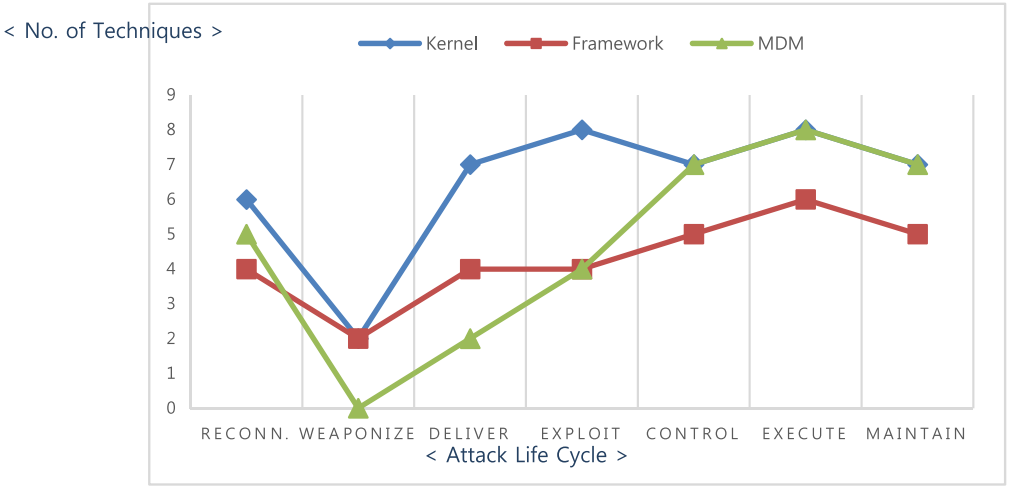
Resiliency techniques in framework		Resiliency techniques implemented in DMOS	
Monitoring and damage assessment	Monitor attack behavior such as attack indication, damage	MDM server monitors asset usage and recognizes invalid behaviors from logs. Response according to the risk levels 1, 2, and 3	F6-DEVMT M8-MONDS
Malware analysis	Analyze malicious code behind adversary activities	Analyze attack behavior by process trace or log audit to detect malware	K10-INTEG F4-APPCH M8-MONDS
Technical defense-in-depth	Use multiple protective mechanisms at many layers or locations	Defend at app-framework-kernel-boot layers, encrypt sensitive data, wipe out data on lost device	K1-SEPOL K9-FENDE F1-SEPOL F5-CERTI
Coordination and consistency analysis	Cyber COAs are defined and executed in coordinated way	Define allow actions definitely by security policy based on RBAC mechanism	K2-FHOOK M4-POLED
Obfuscation	Hide, transform, obfuscate information from adversary	Conceal files related to security. No show target of attackers	K4-CONCL M5-LOGTR
Asset mobility	Physically relocate assets (platforms, mobile devices)	Use specialized mobile phone locate assets in several phones	F5-CERTI M2-DEVMT
Dynamic mapping and profiling	Maintain current information about resource status	Maintain information of device, resource, and operational situation	K5-RESCT F6-DEVMT
Non-persistent information	Refresh information periodically, generate it on demand, delete it when no logger needed	Maintain recent information by generating, modifying, deletion on demand. Delete or hide information which attackers may use	M5-LOGTR M8-MONDS
Non-persistent services	Refresh services periodically, generate them on demand, terminate it after completion	Install, execute only permitted services. Terminates services after usage	K6-PROCT F2-APPCT M1-APPMT
Non-persistent connectivity	Establish connections on demand, terminate them after completion or no use	Allow/deny connections such as WiFi, etc. Terminates lost/stolen device after certain durations	K6-PROCT K7-SYSCT M6-HWONF
Privilege management	Define, assign privileges of users, entities based on criteria	Access control by RBAC policy. Manage privileges for subject to access to object	K2-FHOOK K8-SETID M3-USRMT
Privilege based usage restriction	Define, assign usage restrictions on resources	Allow/deny operations such as read, write, execute, delete which processes request	K2-FHOOK M6-HWONF
Purposing	Ensure resources are used consistent with purposes	Only valid app or valid operation permitted by whitelist can use resources.	K3-FCHCT K5-RESCT
Restriction	Remove unneeded risky functions or connections	Limit operations not permitted by white list. Control process, apps, and resources	K2-FHOOK K6-PROCT F2-APPCT F3-HWPER
Protected backup and restore	Back up information and sw to protect CIA and restore it	MDM back up information, software support encryption and decryption	K9-FENDE M2-DEVMT
Surplus capacity	Maintain extra capacity for information storage, processing	MDM back up information and software supply devices, storage at damage	F5-CERTI M2-DEVMT
Replication	Duplicate information and functions in multilocations	Same user group uses same devices. Each group can share information and functions	F5-CERTI F7-AKFIF M1-APPMT
Predefined segmentation	Define enclaves, segments, resources type based on criteria. They can be protected separately or isolated	Manage resources by privilege and processes based on process isolation	K3-FCHCT K6-PROCT K8-SETID
Integrity/Quality checks	Apply and validate checks of the integrity or information, services, components	Check file integrity during execution periodically and block the changed images	K10-INTEG F4-APPCH
Behavior validation	Validate the behavior of system, device under criteria	Check whether file change, resource access, privilege change and app execution are validate	K3-FCHCT K6-PROCT K7-SYSCT K8-SETID F2-APPCT M7-ACTCH M8-MONDS

understand attack situations and response to them promptly. On the whole, about 15 techniques of DMOS can be applied to cope with each step of the attack, and the most techniques are deployed to defend the execute step. As a result, we can reason that secure kernel techniques are essential for blocking attack.

Table 3 Potential resiliency effects of DMOS on cyber attack life cycle

Attack cycle		Reconn.	Weapon.	Deliver	Exploit	Control	Execute	Maintain
Resiliency technique								
Framework	DMOS							
Monitoring and damage assessment	F6, M8	Detect		Detect		Detect	Analyze	Detect
Malware analysis	K10, F4, M8			Analyze	Analyze	Analyze		Analyze
Technical defense-in-depth	K1, K9, F1, F5		delay		Delay degrade			
Coordination and consistency analysis	K2, M4					Detect delay degrade	Delay degrade	Detect delay degrade
Obfuscation	K4, M5	Delay degrade					Degrade	
Asset mobility	F5, M2	Curtail				Curtail	Curtail	Curtail
Dynamic mapping and profiling	K5, F6					Detect		Detect
Non-persistent information	M5, M8						Curtail	
Non-persistent services	K6, F2, M1				Curtail expunge	Curtail expunge	Curtail	Curtail expunge
Non-persistent connectivity	K6, K7, M6	Delay degrade		Negate		Curtail	Curtail	Curtail
Privilege management	K2, K8, M3	Delay degrade			Contain delay negate	Contain delay negate	Contain delay negate	Contain delay negate
Privilege-based usage restriction	K2, M6				Contain negate degrade	Contain negate degrade	Contain negate degrade	Contain negate degrade
Purposing	K3, K5			Delay degrade	Delay degrade			
Restriction	K2, K6, F2, F3	Delay degrade		Negate degrade		Negate degrade	Negate degrade	Negate degrade
Protected backup and restore	K9, M2						Curtail recover	
Surplus capacity	F5, M2						Degrade recover	
Replication	F5, F7, M1						Degrade recover	
Predefined segmentation	K3, K6, K8	Contain		Degrade		Contain detect delay degrade	Contain detect delay degrade	Contain delay degrade
Integrity/quality checks	K10, F4			Negate detect		Detect	Recover	Detect
Behavior validation	K3, K6, K7, K8, F2, M7, M8					Detect curtail	Detect curtail	Detect curtail
Possible resiliency effects		Detect, delay, degrade, contain, curtail	delay	Detect, analyze, delay, degrade, negate	Analyze, delay, degrade, curtail, contain, negate, expunge,	Detect, analyze, delay, degrade, curtail, contain, negate, expunge	Detect, analyze, delay, degrade, curtail, contain, negate, expunge, recover	Detect, analyze, delay, degrade, curtail, contain, negate, expunge
Supported DMOS techniques		15 K:6, F:4, M:5	4 K:2, F:2	13 K:7, F:4, M:2	16 K:8, F:4, M:4	19 K:7, F:5, M:7	22 K:8, F:6, M:8	19 K:7, F:5, M:7

Next, for the second aspect, we analyze the tendency of secure kernel, framework, and management techniques to support each resiliency effect. Figure 6 shows the result. The most techniques are used to delay and degrade attacks. Relatively, many secure kernel techniques are applied for the detect effect. This is to find attacks in the kernel level as early as possible. The overall tendency shows that secure kernel techniques play an important role for resiliency effects. In order to



Attack DMOS	Reconnai.	Weaponize	Deliver	Exploit	Control	Execute	Maintain	Average Tech./step
Kernel	6	2	7	8	7	8	7	6.4
Framework	4	2	4	4	5	6	5	4.3
Management	5	0	2	4	7	8	7	4.7
Sum	15	4	13	16	19	22	19	15.4

Fig. 5 Techniques of DMOS on attack life cycle

strength resiliency effects, techniques for analyze, contain, expunge, and recover effects must be reinforced.

5.3 Testing cyber resiliency of DMOS

In order to confirm the characteristics analysis above and verify resiliency capability of DMOS, we test whether DMOS techniques can block diverse cyber attacks along attack steps in reality. The testing has executed on the test-bed environment composed of smartphones and a management server. For testing, we devise typical cyber attack scenarios which may happen on mobile phones: S1, S2, S3, and S4 [19, 20]. S4 is the deepest attack scenario. We assume that an attacker has already hunted the security weaknesses of the mobile phone and weaponized the malicious code to exploit.

- Attack scenario S1. Attacker tries to connect to a lost phone using a USB and then to deliver malicious code into the phone in order to extract the information stored in phone.
- Attack scenario S2. Assuming that the USB connection to the phone is satisfied, the attacker delivers certain

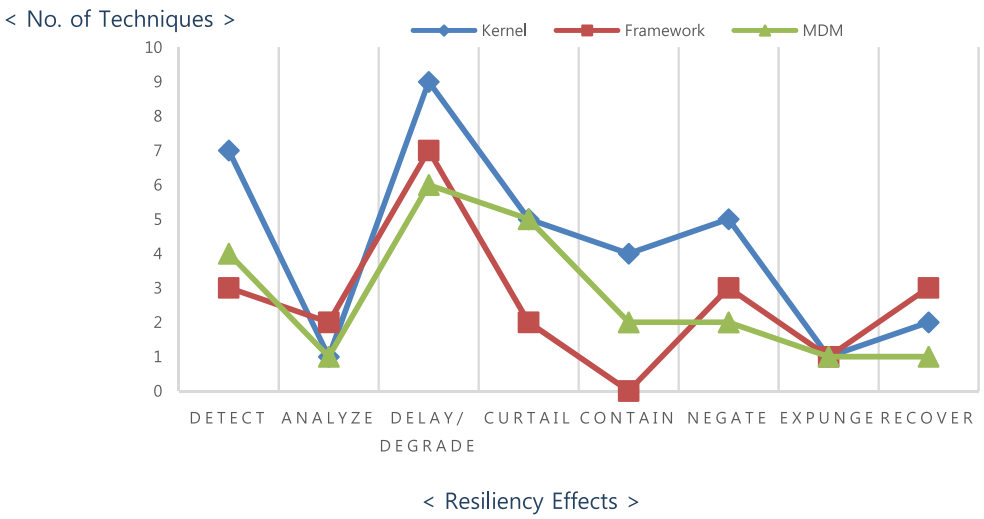


Fig. 6 Techniques of DMOS for resiliency effects

malicious codes into the phone and exploits security weaknesses on the phone. Then, the attacker tries rooting to obtain system privilege.

- Attack scenario S3. Assuming that the rooting is satisfied, the attacker with privilege tries to control exploits on the phone and execute attacks by his laptop in remote.
- Attack scenario S4. After a successful attack, assuming that malicious codes are installed on the phone in secret, the attacker in remote controls the codes to carry out information such as pictures, SNS messages, or data recorded in phone to a C&C server.

For each attack scenario, we test whether attacks can be blocked by DMSO techniques. The test results are as follows.

For scenario S1, with the techniques of DMOS detecting and negating the unpermitted devices connection, the connection trial is failed, the failure message is promptly displayed on phone, and the information about the connection trial and failure is transmitted and logged to the management server. K7, F3, F6, and M8 techniques are primarily used to cope with this attack scenario. They are the techniques for detecting and negating attack in the delivery step.

For scenario S2, by privilege management techniques limiting rooting, the attacker fails to obtain root privilege, the failure message is promptly displayed on phone, and the rooting trial event information is transmitted and logged to the management server. K8, F1, and M8 techniques in the exploit step are applied to response to this attack scenario.

For scenario S3, by using the techniques of DMOS restricting access to sensitive data on phone, the attacker fails to access to data. The result message is promptly displayed on phone, and the event information is transmitted and logged to the management server. We confirm that after a certain time has passed, the rooted phone is initialized in a factory state. K2, K3, K5, F4, and M6 techniques in the control and execute step are the principal techniques to block this attack.

For scenario S4, since privilege control techniques limit the unpermitted app execution and hardware usage, the attacker can't outflow the messages, addresses, and record data in the phone to the C&C server. K2, F2, and M8 techniques in the control step are effectively used to defend against this attack.

Figure 7 shows the attack scenarios along the attack cycle and testing result in short. Although attacks connecting a USB, delivering malicious code, rooting the system, and controlling and executing malicious apps are executed; all computing resources in phone are protected, and missions are operated continuously. Usually, since attacks act with complicated behaviors and intrude through several steps, DMOS defends attacks by more techniques in multiple steps rather than by a one-kill technique within one attack step. By the means of

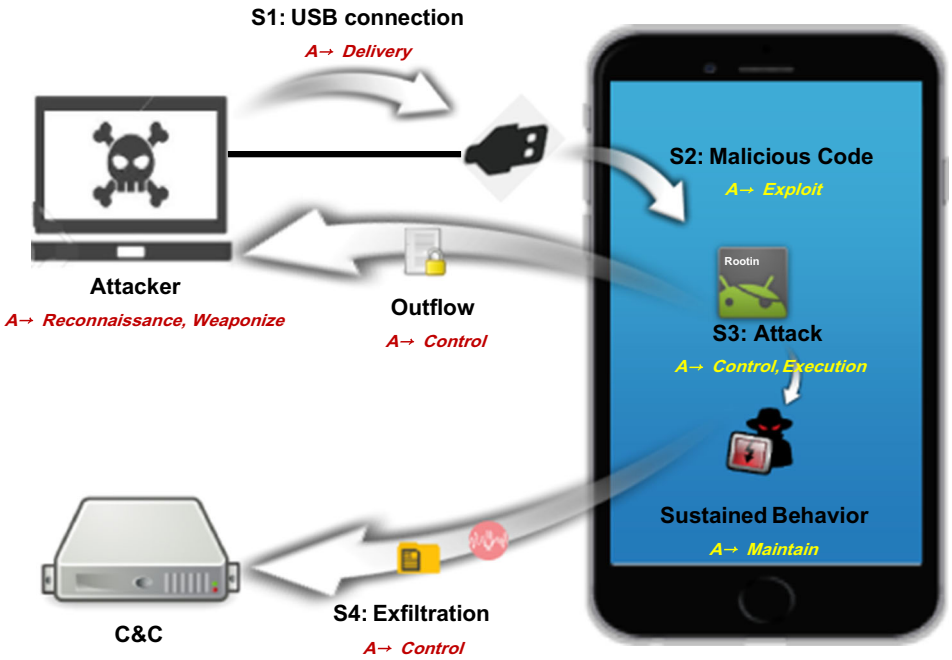


Fig. 7 Cyber resiliency testing scenarios and testing result

Attack scenario	Attack action in scenario	Attack step in attack life cycle	Techniques used in DMOS	Test results
S1	USB connection	Delivery	K7, F3, F6, M8	Attack blocked
S2	Malicious code	Exploit	K8, F1, M8	Attack blocked
S3	Malicious behavior	Control, Execution	K2, K3, K5, F4, M6	Attack blocked
S4	Data exfiltration	Control	K2, F2, M8	Attack blocked

the scenario based on attack defense test above, we verify that the techniques in DMOS can block diverse cyber attacks step by step in a defense-in-depth manner.

In comparison with other OS security techniques, DMOS has a layered security feature. This feature makes it possible to apply security functions in depth and support to enhance security capability in systematical. Also, DMSO can strictly control app execution or resource access in operation level by role-based access control. So, malicious behaviors are blocked even after intrusion into the devices. By the device management technique, we can control services on/off in remote, reason the stolen device and forcibly wipe out data so that they cannot be misused by attacker. For special environment such as small units, we can set the security policies by group and run several classified units. These techniques are newly adapted and reinforced in DMOS.

6 Conclusion

Recently, even the home is being digitalized and automated using IoT techniques or smartphone, etc. However, worries about privacy or economic damages caused by cyber attacks are increasing. Since OS must be used in most computing environments, a secure OS will make computing environments secure. As one of the secure OS techniques, we developed a mobile OS security technique DMOS. It has a deep security ability that blocks attacks in a defense-in-depth manner so that contents are protected and services can be continued even though attacks intrude into devices.

This paper analyzed how well DMOS can realize such deep defense ability. For systematic analysis, referring to the cyber resiliency framework, we identify the techniques of DMOS related to cyber resiliency, examine the resiliency effects which they can provide during the cyber attack life cycle, and analyze the resiliency characteristics of DMOS. For verification, we test the resiliency capabilities of DMOS under real cyber attack scenarios. From such analysis and test results, we find out that DMOS has the techniques able to cope with all steps of the attack life cycle and to provide diverse resiliency effects. Comprehensively, we can assess that DMOS has the potential resiliency capability enough to defend against cyber attacks. With its resiliency capability, DMOS can provide deep security ability for secure personal ubiquitous computing.

The DMOS technique may be applied to protect the devices, systems, or services operated on the Android OS. The approach assessing defense capability in the aspect of cyber resiliency can be a useful way to check the security ability. For further study, the cyber resiliency assessment in real environment, cyber resiliency metrics, and other advanced resiliency techniques must be studied [21–24]. Also research on deep security is needed.

References

1. T. Guo, P. Zhang, H. Liang, S. Shao (2013) Enforcing multiple security policies for android system, In: Proc. of the 2nd International Symposium on Computer, Communication, Control and Automation, pp. 165–169
2. Draft version 1.1 (2017) Framework for improving critical infrastructure cybersecurity, National Institute of Standards and Technology, pp. 1–57, Jan.
3. Mobile security R&D program guide, Homeland Security, vol. 1, pp. 1–48
4. D.J. Bodeau, R.D. Graubart, E.R. Laberman (2014) Cyber resiliency engineering overview of the architectural assessment process, In: Proc. of Conference on Systems Engineering Research (CSER2014), pp. 838–847
5. ADD (2016) Development specifications for mobile OS security, http://www.add.re.kr
6. Lee S, Kang T (2015) Adaptive multi-layer security approach for cyber defense. J Internet Comput Serv (JICS) 16(5):1–9
7. C. Williams, T. Watson et al (2012) Resilient cyber ecosystems, Crosstalk Journal of Defense Software Engineering, vol.25, no.5, US Air Force, Sep./Oct
8. G. Jakobson (2013) Mission-Centricity in cyber security: architecting cyber attack resilient missions, In: Proc. of the 5th International Conference on Cyber Conflict, pp. 1–18
9. Goldman HG (2010) Building secure, resilient architectures for cyber mission assurance. MITRE Technical Report, pp.:1–18
10. J.B. Rice Jr, F. Caniato (2003) Building a secure and resilient supply network, Supply Chain Management Review, Sep/Oct. pp. 22–30
11. J. Snyder (2006) Six strategies for defense-in-depth, OPUS, pp. 1–9
12. K. Cox, D. Bodeau, R. Graubart (2015) The cyber resiliency framework: planning for cyber attack survival, MITRE presentation pp. 1–55
13. S. Wagner, E. van den Berg, J. Giacopelli, P. Manghwani (2012) Autonomous, collaborative control for resilient cyber defense (ACCORD), In: Proc. of IEEE 6th International conference on Self-Adaptive and Self-organizing Systems Workshops, pp. 39–46
14. Florio VD (2014) Antifragility = elasticity + resilience + machine learning models and algorithms for open system fidelity. Proc Comput Sci 32:834–841
15. D. Bodeau, R. Graubart, J. Picciotto, R. McQuaid (2011) Cyber resiliency engineering framework, MITRE Technical Report 1–68
16. D. Bodeau, R. Graubart, W. Heinbockel, E. Laderman (2015) Cyber resiliency engineering aid-the updated cyber resiliency engineering framework and guidance on applying cyber resiliency techniques, MITRE, 1–63
17. Velazquez C (2015) Detecting and preventing attacks earlier in the kill chain. SANS Institute Infosec Reading Room, pp.:1–21
18. D. Bodeau, R. Graubart, L. LaPadula, P. Kertzner, A. Rosenthal, J. Brennan (2012) Cyber resiliency metrics, version 1.0, rev. 1, MITRE Technical Report, pp. 1–34
19. P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M.S. Gaur, M. Conti, M. Rajarajan (2015) Android security: a survey of issues, malware penetration and defense, IEEE Communications Surveys and Tutorials 1–27
20. Sanghvi HP, Dahiya MS (2013) Cyber reconnaissance: an alarm before cyber attack. Int J Comput Appl 63(6):36–38
21. B. Schmerl, J. Camara, J. Gennari, D. Garlan, P.Casanova, G.A. Moreno, T.J. Glazier, J.M. Barnes (2014) Architecture based self-protection: composing and reasoning about denial-of-service mitigations, In: Proc. of the 2014 Symposium and Bootcamp on the Science of Security, pp. 1–12
22. E. Yuan, S. Malek (2012) A taxonomy and survey of self-protecting software systems, In: Proc. of the 7th International Symposium on

Software Engineering for Adaptive and Self-Managing Systems, pp. 109–118

23. J. Newsome, D. Brumley, D. Song (2005) Sting: an end-to-end self-healing system for defending against zero-day worm attacks on commodity software, Carnegie Mellon University pp. 1–27

24. P. Ramuhalli, M. Halappanavar, J. Coble, M. Dixit (2013) Towards a theory of autonomous reconstitution of compromised cyber-systems, In: Proc. of International Conference: Technologies for Homeland Security (HST), pp. 577–583