International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 3 Issue 4, April – 2019, Pages: 9-13

Handles for Pentesting
Modern Secure Coding: bypassing mobile security

Mourad M.H Henchiri

Lecturer: dept. of Information Systems, University of Nizwa, CEMIS, Nizwa, Oman
mourad@unizwa.edu.om

Abstract— Malware behavior was and still is a key solution, for top security appliances, to monitor algorithmic approaches when performing regular security tasks; scan, detection, cleaning and removal. And even for early actions; when building a security framework and securing all possible access points to all data sources. The first suspect in such scenario is the inner residents; appliances and system functions. Numerous are available at each operating system, and thus, the security is raised and set up frequently with all nowadays, yet, still we are able to identify black wholes and back doors, even with the high security approaches applied at different levels and layers. In this research we are presenting an inevitable security thread, we face frequently in different scenarios, that are the key definition for successful security trespass, which we secure with an ameliorated security shell skeleton. Though the sysinternals are doing sufficient services to help administrators accomplish security tasks; automating administration within large scale environments.

Keywords— pentest, thread, sysinternal, shell skeleton**.**

1. Introduction

It is a current necessity today; securing digital environments from all unwanted data leakages. Operating systems developers and secure coding engineers are frequently developing solutions along with the care of security perspectives, yet, all customizable; designated to public and commercial use, which is a scenario where the security is manifested in its weakest state. Here, in such public environments, communication channels between objects; threads, processes and APIs are exposed and seen clearly; handles of each running process is the deepest place where malware act; hide processes, replace a true process with a fake infected one, delete threads… Penetration testing is a vitality to a digital environment, in the sense of securing all communication channels. This security has to be in parallel to each data flow and activity without causing time latency to any. Yet, script kiddies are able to break these rules since simple algorithms manifested in a script written in VBA or JS or SHELL or any other scripting or programming language that has the capabilities to launch operating system commands and execute orders to break up validated handles and generates fake entries that a system has to trust based on the handled software authentication. Even new handles can be created and linked to injected utilities running on the OS [5, 6]. And all the new handles are to be an element in a linked list that the OS truly trusts and gives it the current user privileges, and in return the running malware with its handles would act accordingly with equal wrights and privileges and even may raise its rights to a high scope or make privilege escalation to a super user rights. All to be done while the OS is trusting such behavior. Such a behavior is usually persistent, remains active and effective for long times and infects all encountered processes and threads. Even most expert and skilled security technicals find difficulties in judging whether a process or its handle are fake from a first view. Malware behavior is a smart act that lures the reader and the user to achieve negative realizations and compromises resources. The point here is not in to list the processes but identify malware processes, and what we have proven in this research, we will present it within the research contribution, is that malware are smart enough today to hide them selves; because of the sophisticated techniques applied. So, detecting malware is much easier than extracting hidden malware; which are not even listed as a process in the OS. Thus, in this research we are to prove hidden malware capabilities and persistence in doing bad acts and generating back doors. Also, we would be giving birth to a shell skeleton to automate administrative tasks versus security.

The sophistication of the attacks here, within the research, is applied over mobile devices; attacking android-based devices via a standard communication channels, to succeed injecting the malware then activating and executing its payload [16].

This combined view; the system handles and the networks attacks on the android-based devices is the spark that pushed this research to life. Thus, this study in a clearer view is going to define different matters:

- System handles.
- Network attacks.
- Mobile attacks.
- Networks and devices Pen-testing.

A deployed scenario is enhancing the matters appraised.

2. System handles

Persistent processes gain their status due to running handles; which is seen active even after a process is killed. Handles are an OS feature that takes care of running entities from every process. Thus, here, handles prove they are a main actor in the scenario of data communication between parties; processes, threads, system functions and procedures.

A system dedicated path to processes that are active and available in background of all user APIs. As per the chosen name to this feature, "*handle"*, it is the technical handover to every running behavior within the processor; software applications, appliances, utilities, tools, etc… Also, handles are not to be exposed to end-users, yet, they are to be explored when needed; special tools may help retrieve their availability to the light, or special activities over Powershell may prove their persistence to the system.

Handles are a system feature for system behavior, and, for dedicated professional and skilled activities; to build and provide a step further in ameliorating the environment architecture. Yet, and since they are explorable and possibly fetched, they are a weakness and a system vulnerability. Handles, identify processes with their IDs (Process ID), which, in this research, we proven they can be falsified and hidden.

3. Penetration testing

Security deployment forces security specialists to be alerted and ready technically for the up to date technology. Awareness is the first requirement an ethical hacker would acquire to act and be admitted within a security team in an organization. Awareness of all innovations that the Human has achieved in the digital environment within an international scope.

Terminologies in digital security provides a deep learning with trusted skills at first encounter with a real case. It is a trivial dilemma to go after the awareness and data collection defining the wide range of terminologies presented within digital security scenarios. Processes' handles are a key and a pillar solution to follow when doing pen-testing. Thus, penetration testing secrets are revealed here in this study; with the wide scope of the spread channels, the technical skills required by a penetration tester has to be narrowed and dedicated. Strong technical skills used in penetration testing is built behind a strong awareness.

The application of the handles in the scenarios of hacking and cracking, stands brightly as a powerful engine that helps and guides in doing so. Hiding processes and different behaviors within the OS is the most important act malware would do and need. Also, pen-testers, would check security adoption and vulnerabilities availability by going after interrogating processes' handles. These interrogations would be successful at a high rate when good social skills are mastered and deployed. In this research we are willing to generate a fake image to share with our targeted victims; mobiles, and is said fake because it is to be read as an image by all filters and intermediary devices and even to be received as an image by the end node. Yet, the original file is a script file, holding scripts that when executed, trespass the limits by exploring and triggering a payload, generated and saved as an image. For the ease of explanation only image files are discussed, at the time zipped files are a very successful approach to succeed in the same scenario testing security levels and making penetrations to different devices and environments.

Thus, the concern of penetration is achieved due to the limitations in the available technology, here, we are presenting the protocols adopted upon the social media networks to help providing variety of services.

4. Processes, Threads and Parallel Programming

The parallel programming is a good context to which a clever attack can look after to hide attacks; two or more seemingly alike processes are executed at the same time. Our concern here, is to expose the hidden threads along with their silent behaviors. Procedures are called and executed simultaneously causing time gains and faster computations. Yet, here, the security issue is exposed; faster parallel computation means higher risks when controlling activities and computed processes. Then lower chances stop running hidden malware. At the same time, it is to confirm that hidden malware; persistent threats, are not successful due to parallel computations, but, due to their capabilities to modify system handles links and registry keys and even hives [3].

5. Android Mobile's Files Architecture

Penetration testing is a digital security process seeking for security breaches and presenting best reports describing explored weaknesses and estimated trust levels, some scenarios within their reports suggest best solution to adopt in order to strengthen security architecture in question.

Remotely is the first scenario to draw when doing pen-testing; external tracing, scanning, intruding, interrogating, etc… is the first testing plan, which is a concern through this research; we are presenting, over the contribution, social skills and technicalities successfully applied to gain access [1, 15].

A hack, here, is a full and complete scenario deployed; starting from selecting and electing victims, passing by applying technical skills adequate to the environment for the end of gaining a penetration to victims' environment; hardware and software. Where a major act is to be activated and set to achieve the aim and succeed in the hack. Social media networks are a guarantee of success to such penetration. Two main environments are the drive of this study; android based devices and iOS based devices. Shell is a target to be run over each of the platforms. Thus, the real achievement is to attain the execution of the customized payloads estimated to help in compromising the hosting device. The Steganography is our solution to infect every device with a backdoor. Yet, here again, social engineering skills have to take place; in order, and in a second phase, after the Steganography, to extract and execute the hidden payload[4, 7, 16].

Social media network, Steganography skills and mastering social engineering tactics is all what a pen-tester needs to compromise a mobile device. Here, an important consideration has to be raised which is the activities logs; the traces of the attacks and the pen-test processes. A primordial behavior has to be the all concern of the doer; to be hidden and anonymous upon all phases of the attack, thus, the hidden processes would be the first setting to all automations. Because, such a pen-test has to be automated and executed by bots [15].

At this level, the smart device, has no chance escaping the attack, how ever the file's system architecture is. The open social media providing the instant messaging services are a fail story to the digital security in this regard of pen-testing scenarios. Comes in secondary level, the in OS security set on the mobile device, which is not the concern of this research nor the case study [10, 11].

6. Algorithmic Complexity

The scenario applied to achieve the pen-test starts at the level of creating the malware acting anonymously and in hidden mode; means it run with no handle the system processes IDs.

Then processes are to be sequentially as follows:

1. Compromising the victims' devices.
2. Social engineering tactics to deploy the payloads.

7. Background

Clearly, the smart phones' OSs (Android and iOS) have taken over the noticeable market of other old manufacturers such the Symbian and the Blackberry Phones. And since the early era of smart phones up to nowadays, Symbian and Blackberry phones supported the social media networks' services [4, 7, 16].

From the early age the different smart devices supported services over the IM (Instant Messaging) protocol. Consequently this is a vulnerability that started with the technology birth and still it persists.

Addressing smart devices penetration is a target of commercial and economical parties; pushing customized advertisements randomly with a certainty that it would achieve a 100% of reach. At the time, this is a positive realization yet, technical concern pushed the work towards checking vulnerable access points [8].

8. Implementation and Methodology

Modern smart electronic devices come with a default setting having the social media networks utilities installed in prior and ready for usage. In addition to that they are able to compile and interpret a variety of scripting languages, and this gives a flexibilities to malware to run and be executed with a higher level of success. Our first view here, goes intentionally to the shell scripting language, the JavaScript, the Visual Basic for Applications (VBA), and much more. All of the scripting languages are to be specified upon needs and targets of the attack [9].

Platforms like Android supports third party security appliances and Applications. Security services allows controlling which services to run on the device. The handset soft application can request for periodic update of the security appliances. The security appliances can also control the Internet access, but, watermarked files shared withing a data network, is not to be detected or blocked by a security appliance installed and set over a smart handset. One of the main reasons behind, is that they are given the authorization and manually accepted [4, 7].

1. Steganography [13]

   Here we define the different scenario to inject the bot's script in the target file to share, to achieve the scenario a goal has to be set as outcome, to our scenario the followings interfere:

- Bot's script: Based on the OS; Android and iOS, shell scripts are adequate to run and be a backdoor at each when deployed [4].
- Social media networks: This is the super success manifested when all injected or watermarked files are shared and communicated successfully.
- Social engineering: This is the best approach to tackle the victims and deploy the injected and communicated files.

- Behavior automation and anonymous identity: The handles defined by the OS to link and point each running process is the key to create a back door with a hidden identity; no visible handle to our running back door.

2. Social engineering

   The deployment of the malware shared and communicated with the victims has to be deployed as his final state, to have the capability of running and doing the necessary tasks. Thus social engineering is a master skill to succeeding doing so.

   The required output of the social engineering attack are two:

   a) Extract injected script when doing the steganography.

   b) Saving extracted script - Allows the request to have its desired state.

3. Handles

   Hidden the operating system handles makes a process hidden from the view, yet, available to specific search criterion. The automation of the steganography is to be a hidden process within the compromised bots. Then the communications of the injected files is to be also launched by the hidden processes.

9. Conclusion

Pen-testing are tunned up nowadays, and are of a great concern to all technology geeks. The realization of this research is an enhancement to the availability; exposing the smart behavior of malware; being hidden and persistent in background is a primordial behavior after a successful injection and compromising a network device. An attack on a mobile device is discussed with a full scenario describing the different platforms from two sides; vulnerable points and default security perspectives. Life is a joy to all end users of technology manifested in mobile smart handsets, yet, this is vulnerable to a successful attack tactic based on social engineering skills. The implementation of such technology faces a variety of constraints. Which include [1]:

- Technology Constraints

  The next stage of the implementation today is that it has to face compatibility concerns, mapping under the mobile operating systems, which needs to be more comprehensive than how it is definite today. This brings into consideration significant challenges for ameliorating Instant Messaging protocols schema. And all the social media networks has to be seen differently, with a security perspective, parental control like scenarios has to be adopted intentionally, for the safe and sane seek of all users. In developing countries like Tunisia and Oman, the digital technology is in very nascent stage as per the year 2018 in addition to the clear domination of the social culture and faith beliefs, which implies an easier adoption of the stated intentional control over the social media networks across the country.

- Infrastructure Constraints

  Technology is all about the digital communication, yet, and due to the wide variety of technology providers, the interoperability is a question that must be tackled through the protocols activity and technical behavior.

- Market failure

  Commercialization is a target of technology providers thus, rules break up might seen frequently, even with the possible annotations and guidance, thus, here governmental control also is advised to set the quality of services (QoS) headlines.

References

[1] A. G. Bacudio et al, "An overview of penetration testing,"

International Journal of Network Security & its
Applications, vol. 3, pp. 19, 2011.

[2] W. G. Halfond, S. R. Choudhary and A. Orso, "Penetration testing with improved input vector identification," in 2009 International Conference on Software Testing Verification and Validation, 2009, pp. 346 – 355.

[3] J. Hoffmann et al, "SAT encodings of state - space reachability problems in numeric domains." in Ijcai, 2007, pp. 1918 – 1923.

[4] GPS and Localization Web Services Implementation over Android Modern Satellite Navigation

International Journal of Engineering and Information Systems (IJEAIS), 2017, 1 (8), pp.220 – 229

[5] I. Arce and G. McGraw, "Guest editors' introduction: Why attacking systems is a good idea," IEEE Security & Privacy, vol. 2, pp. 17 - 19, 2004.

[6] A. E. Gerevini, A. Saetti and I. Serina, "An approach to efficient planning with numerical fluents and multi-criteria plan quality," Artif. Intell., vol. 172, pp. 899 - 944, 2008.

[7] P. Halsum and H. Geffner, "Heuristic planning with time and resource," in IJCAI Workshop on Planning with Resources, Seattle, USA, 2001, .

[8] M. Steinmetz, "Critical constrained planning and an application to network penetration testing," in The 26th International Conference on Automated Planning and Scheduling, 2016, pp. 141.

[9] J. Hoffmann and M. Fickert, Explicit Conjunctions W/O Compilation: Computing hFF (Πc) in Polynomial Time (Technical
Report), 2015.

[10] P. P. Shimpi and M. S. Nagpure, "Decentralized Virtual VAPT Laboratory Model," Global Journal for Research Analysis, vol. 4, 2016.

[11] L. Vishnoi and V. Shrivastava, "Results of Penetration Testing on Various Operating Systems," .

[12] (2001). Penetration Tests: The Baseline For Effective Information Protection Available: http://www.iss.net/documents/whitepapers/pentestwp.pdf.
DOI: 14 – 12 – 2016.

[13] (2014). Penetration Testing. Available: https://www.depts.ttu.edu/cs/research/csecs/workshop/docs/.../PetetrationTesting.ppt. DOI: 14 – 12 – 2016.

[14] Location the Portal on positioning and navigation
www.location.net.in

[15] LBS Zone www.lbszone.com

[16] Android Wireless Application Development

By Shane Condor and Lauren Darcy

http://ptgmedia.pearsoncmg.com/images/9780321947864/samplepages/032194786X.pdf