BLACK CELL
Protecting critical infrastructures

# Offensive Security Whitepaper

# Table of Contents

# 1. ICS/SCADA vulnerability testing

The utilized testing method is based on international standards (NIST SP 800-15) combined with Black Cell's own methodology. Built on these steps, the test is performed to analyze the ICS system and its infrastructure to identify as many vulnerabilities as possible within the defined testing time.

The main purpose of the test is to ensure the continuous production and operation of the system, therefore, if it is possible, the test is conducted in a test environment. If providing a testing environment is not viable, the testing is run strictly controlled and supervised. During the first step, network discovery is run to map the tools, applications and devices available on the network. Network and endpoint protection tools are also reviewed and checked for known vulnerabilities. In the second step, we examine the services, protocols used by the devices and check which devices can communicate with others. We also test PLC-s, RTU-s, ICS-specific routers and switches, HMI and other devices for known vulnerabilities either in Application and OS Layer. If exploitation of these vulnerabilities does not interfere the system, we also validate the findings and create risk assessment report for the organization.

The result of the process is a summary report, which contains a detailed description of the procedure, methodologies and tools used, the vulnerabilities discovered, their risk classification and assessment, and the recommended remediation steps for their correction.

The main steps of ICS/SCADA vulnerability testing

- **Discovery:** the purpose of this step is to examine network separation, review the workings of network and endpoint protection solutions. During the testing, we also check whether these devices and services can be accessed from external sources and if they are exposed to the Internet, check their presence in online databases.
- **Availability check:** during the testing phase, we check the availability of internal network devices and services: how they communicate, which protocols are being used, etc. We also test password usage, protection of communication channels, outdated VPN and other management programs and look for vulnerable points in the system. It is also checked, whether the network can be accessed through the WiFi network. Finally, the cryptographic functions are checked, whether the protocols have known vulnerabilities and outdated instances, which would not be accepted by the NIST standard anymore
- **Testing access of the devices:** the purpose of this test is to map the inbound and outbound firewall rules and communication of the devices, check network connection (e.g. whether they are connected to the Internet), the communication of the devices with other network segments (encrypted or unencrypted channel, which ports and protocols are being used) and look for known vulnerabilities.
  Analyzed tools: RTU (Remote Terminal Unit), PLC (Programmable Logic Controller), IED (Intelligent Electronic Device), ICS controller (HMI – Human-Machine Interface), ICS specific routers, switches, gateways, protocol converters, databases and other tools (e.g. thermometer, sensors, IIoT tools).
- **Test for OS and Application Layer vulnerabilities:** During the testing phase, we analyze the environment of the devices (e.g. which OS they are running on, do they have the proper patches installed, appropriate policies are being used, is the configuration sufficient). We also check whether the services and protocols used by these devices and applications have known vulnerabilities, do they have proper authentication and authorization mechanism, can sensitive data be extracted from them or is it possible to

sniff on the network to gain and modify data sent on these channels. The system-critical vulnerabilities are validated only in testing environment.

Tools and utilities used for the testing

- Automatic vulnerability scanners (e.g. Nessus, Nexpose)
- Special discovery tools (e.g. GRASSMARLIN)
- Special PLC scanners (e.g. s7scan, plcscan, ModScan, mbtget, smod)
- Special RTU analyzers (e.g. sixnet-tools)
- Port scanners (e.g. nmap, masscan)
- Network sniffing tools (e.g. WireShark, BurpSuite Professional, tcpdump)
- OSINT tools (pl. Shodan, Spiderfoot)
- Proprietary scripts and applications.

## 2. Internal network penetration testing

The utilized testing method is based on international standards (e.g. OWASP, NIST SP 800-15, OSSTMM) combined with Black Cell Ltd.'s own methodology. Built on these steps, the test is performed to analyze the application and its infrastructure to identify as many vulnerabilities as possible within the defined testing time.

The test begins with an analysis of the structure of the internal network. Prior to testing, it is discussed with the Client, which part of the internal network can be assessed, which devices and services are critical points of the system and what kind of access is provided for the testing team (e.g. on-site audit is needed or VPN is provided). The next step of the testing is to map the internal network and provide its topology. The vulnerabilities are identified either by automatic scanners and manual scripts and they are detected both in application and OS layer. The collected data is then analyzed and validated manually to filter the false positive results out. In the risk assessment phase, the gathered information is classified. It is outlined, what are the risks and the consequences of a successful attack. The result of the process is a summary report, which contains a detailed description of the procedure, methodologies and tools used, the vulnerabilities discovered, their risk classification and assessment, and the recommended remediation steps for their correction.

The main steps of internal network vulnerability testing:

- **Information gathering:** in this phase, it is discussed with the Client, what kind of access is provided for the testing team (e.g. on-site audit is needed or VPN is provided) and which points of the system are critical and to be tested within certain limits. It is discussed, which part of the internal network should be tested: which segments (IP address ranges, applications, endpoints) should be audited and which tools can be used not to interfere the workings of the internal system.
- **Network scanning:** in this phase, the segments, network devices and their versions, services, communication protocols are identified and then a network topology is created for the latter part of the audit.
- **Network vulnerability scanning:** in this phase, either automatic vulnerability scanners and manual tests are utilized to discover the protocols, services, their version numbers and the corresponding vulnerabilities. Network segmentation, endpoint and network protection tools (e.g. firewalls, antivirus solutions, IDS/IPS) are checked whether they are working properly. The testing includes assessing the security maturity of cryptographic protocols and available configuration settings.
- **OS and Application Layer security testing:** during the testing, we analyze the permissions, policies of the users of each individual OS. We determine the running services, settings and version numbers of each application, then we search for points that are possibly vulnerable. The process includes identifying vulnerabilities, testing authentication and error handling.
- **Logs and configuration review (optional):** during the testing phase, we audit the configuration settings of the applications and devices, check whether the sensitive information is handled and stored correctly. It is also tested, whether the logs are set and processed properly.
- **Testing the WiFi network (optional):** the testing phase consists of the audit of the access points and searching for specific vulnerabilities in the devices, which could be

exploited to gain access for the internal network. The communication between the access points and other devices is also tested in order to find further vulnerabilities.

Tools and utilities used for the testing

- Automatic vulnerability scanners (pl.Nessus, Nexpose, InsightVM)
- Port scanners (pl. nmap, masscan)
- Network sniffing tools (pl. Wireshark, BurpSuite Professional)
- Exploitation tools (Metasploit Pro, Hydra)
- WiFi és Bluetooth hacking tools (pl. Alfa, Ubertooth)
- Proprietary scripts and applications.

# 3. Mobile application vulnerability testing

The main purpose of the Mobile Application Vulnerability Testing is to determine the maturity level of information security of the application and discover vulnerabilities from code-level bugs to improper configuration, which can lead to data infiltration and compromise of the application.

During the investigation, the vulnerabilities are identified and validated, risk assessment is processed and the impact of a potential attack is analyzed. The utilized testing method is based on an international standard (OWASP Mobile Security Testing Guide) combined with Black Cell's own methodology. We analyze the impacted Android and IOS applications to discover and validate as many vulnerabilities as possible within the defined testing time.

The main steps of mobile application vulnerability testing

- **Information gathering/OSINT:** the testing consists of extracting and analyzing metadata from files, finding users, version numbers and vulnerable components using OSINT tools.
- **Local storage analysis:** the purpose of the step is to check the configuration settings of the database, analyze XML, SQLite, secret key files and search for misconfigured instances, which may be available from external sources or stored unencrypted.
- **Sensitive information handling**: the main purpose of the testing is to analyze the data stored by the application and test whether the sensitive information given by user is stored and processed (setting logs, information given to third parties, IPC communication) properly.
- **Authentication testing:** the purpose of this phase is to determine whether the authentication scheme can be bypassed by an attacker and whether the lockout mechanism is properly working, or is it possible to retrieve and reuse the users' or administrators' login information. The analysis of the session tokens and cookies is also scope of this testing phase.
- **Input validation testing:** during the testing, various user inputs are given to the application. By these requests, the input validation mechanism of the application is tested. The aim of the test is to verify whether the application or the server allows the attacker to bypass the input validation mechanism (if it presents) and launch different forms of attacks (SQL Injection, XML Injection, Command Injection, Code Injection, File Inclusion).
- **Network communication testing:** during this phase, the network communication is examined and tested for man-in-the-middle attacks to identify vulnerabilities in the implementation or in the communication protocol, which can lead to data infiltration. The test also contains the analysis of the certificates.
- **Cryptography testing:** the purpose of the testing is to find channels where data is sent unencrypted and to identify whether cryptographically strong algorithms are forced to be used. It also determines whether the used algorithms are supported by NIST or whether they have known vulnerabilities in their protocol, which is usually exploited by attackers to mount further attacks.
- **Reverse engineering**: the aim of the investigation is to find out, whether the application has been implemented using proper anti-reverse engineering methods (e.g. obfuscation, encryption) and try to extract data while reversing the application.

- **Rooted and jailbreaked device detection:** the aim of the analysis is to analyze whether the application properly identifies the rooted or jailbreaked devices.

Tools and utilities used for the testing

- Reverse engineering tools (e.g. IDA, apktool, Frida, Hopper)
- Decrypter/decoding tools (e.g. Clutch, Class-Dump-Z)
- Network sniffing tools (e.g. Wireshark, BurpSuite Professional)
- OSINT tools (e.g. Shodan, Spiderfoot)
- Proprietary scripts and applications

# 4. OSINT (Open Source Intelligence) investigation

The utilized testing method is based on the OSINT section of PTES international testing guide. The test is performed to analyze the data of the organization that can be collected from open source and publicly available sources. The main goal of the OSINT method is to collect all available data and analyze it to produce organization-specific information on attack surface, weaknesses and the level of security awareness of the workers to review the security posture of the company.

The first step of the OSINT investigation is to define the target and type of the analysis, which is preliminary consulted with the organization. The next step is to acquire data from publicly available sources and then analyze it to find relationships, weak points that can lead to attack surfaces or other relevant information. Based on the data collected, the steps among are enumerated by defining new targets. The last step of the investigation is to summarize the findings in a report which contains a detailed description of the procedure, methodologies and tools used, the data collected, the identified weaknesses and the recommended remediation steps to increase the security posture of the organization.

The main steps of OSINT investigation:

- **Target selection:** during this step, the experts consult with the organization and identify the target of the investigation. They discuss the rules, limitations, the length and the end goal of the analysis. The investigation can happen in more forms; it is possible to conduct Passive Information Gathering, Semi-passive Information Gathering and Active Information Gathering, which are different in their method and detection rate.
- **Corporate information gathering:** during the phase, information is collected about the location and relationships of the organization. It is aimed to identify business partners, clients, competitors, marketing accounts, significant company records and job openings, which can help to map company infrastructure (e.g.: used technology, number of workers, communication lines, external users and the applied policies, restrictions, cybersecurity solutions).
- **Digital information gathering:** the aim of the test is to identify the digital environment of the targeted organization by analyzing IP ranges, email addresses, technology, applications and defense technology. During the test, DNS records, certificates, IP ranges, subdomains and metadata are analyzed to this information. The final part of the step is to identify open ports and running services on the corporate website. This part can also be offline and online depending on the agreement between the organization and Black Cell Ltd.
- **Individual information gathering:** the main goal of the phase is to identify employees of the organization. Data is collected from social network profiles, professional services and the experts try to find common points with the corporate network (e.g. username or password reuse, patterns). Breach databases are investigated to find valid credentials, that could be used for a targeted attack.
- **Archived information analysis:** during the test, archived resources are being analyzed. This information is usually no longer available on live websites however, it is possible to retrieve them by using special tools and sites. It is investigated whether the available information can be used to find vulnerable points of the organization (e.g. passwords used for the old database, usernames, programs, domains).

Tools and utilities used for the testing

- Port scanners/analyzers (e.g. nmap, masscan, Shodan)
- Metadata analyzers (e.g. FOCA, Maltego)
- Archive information gathering tools (e.g. archive.org, GHDB searches)
- Breach database searches
- Social media searches
- Proprietary scripts and applications.

# 5. Software vulnerability testing

The aim of security testing „Thick Client" applications and software is to identify the vulnerabilities of the installed application, review its configuration settings and analyze network communication.

During the testing the vulnerabilities are identified and validated. We also assess the risks of each point, and analyze the consequences of a potential attack. Throughout the testing, Black Cell uses its own methodology to investigate the security posture of the mentioned applications to determine and validate as many vulnerabilities as possible within the defined testing period.

The investigation starts with an information gathering phase (from OSINT sources) to identify vulnerable components, keys and users. Next, we analyze the network communication, cryptographic protocols and authentication mechanism. We try to reverse the code of the program by using various reverse engineering and debugging tools to extract sensitive information such as users and secret keys. Moreover, it is tested, whether the program contains parts that have known software vulnerabilities (buffer overflow, unused variables). The collected data is then analyzed and validated manually to filter the false positive results out. In the risk assessment phase, the gathered information is classified. It is outlined, what are the risks and the consequences of a successful attack. The result of the process is a summary report, which contains a detailed description of the procedure, methodologies and tools used, the vulnerabilities discovered, their risk classification and assessment, and the recommended remediation steps for their correction.

The main steps of the „Thick Client" application vulnerability testing:

- **Information gathering/OSINT:** the aim of the testing is to extract and analyze metadata from files, find users, version numbers and vulnerable components, using OSINT tools.
- **Local storage testing:** the purpose of the test is to analyze the method of saving data on the local storage and search for not properly stored information (e.g. unencrypted data, information available from external sources).
- **Sensitive information handling**: the aim of the testing is to analyze, whether sensitive data -provided by the user- is stored and processed properly (e.g. log information, data provided for third parties) by the application.
- **Authentication testing:** the purpose of this phase is to determine whether the authentication scheme can be bypassed by an attacker and whether the lockout mechanism is properly working, or is it possible to retrieve and reuse the users' or administrators' login information.
- **Input validation testing:** during the testing, various user inputs are given to the application. By these requests, the input validation mechanism of the application is tested. The aim of the test is to verify whether the application or the server allows the attacker to bypass the input validation mechanism (if it presents) and launch different forms of attacks (SQL Injection, XML Injection, Command Injection, Code Injection).
- **Network communication testing:** during this phase the network communication is examined and tested for man-in-the-middle attacks to identify vulnerabilities in the implementation or in the communication protocol, which can lead to data infiltration. The test also contains the analysis of the certificates.

- **Cryptography testing:** the purpose of the testing is to find channels where data is sent unencrypted and to identify whether cryptographically strong algorithms are forced to be used. It also determines whether the used algorithms are supported by NIST or whether they have known vulnerabilities in their protocol, which is usually exploited by attackers to mount further attacks.
- **Reverse engineering:** the purpose of this testing phase is to check whether the application uses proper obfuscation or is it possible to reverse the application to extract sensitive information.

Tools and utilities used for the testing

- Reverse engineering tools (e.g. IDA, Immunity Debugger, DnSpy, Java Decompiler)
- Sniffing tools (e.g. WireShark, BurpSuite Professional)
- OSINT tools (e.g. Shodan, Spiderfoot)
- Proprietary scripts and applications.

# 6. Web application vulnerability testing

The utilized testing method is based on international standards (e.g. OWASP, NIST SP 800-15, OSSTMM) combined with Black Cell Ltd.'s own methodology. Built on these steps, the test is performed to analyze the application and its infrastructure to identify as many vulnerabilities as possible within the defined testing time.

The investigation begins with a discovery phase, gathering data from publicly available sources (OSINT – Open Source Intelligence) and evaluate the information found on the target. Then the next process begins with detailed scanning and research into the architecture and environment, with the performance of automated testing for known vulnerabilities. It is followed by a manual exploration of the vulnerabilities, for the purpose of detecting security weaknesses in the networks in scope. The collected data is then analyzed and validated manually to filter the false positive results out. In the risk assessment phase, the gathered information is classified. It is outlined, what are the risks and the consequences of a successful attack. The result of the process is a summary report, which contains a detailed description of the procedure, methodologies and tools used, the vulnerabilities discovered, their risk classification and assessment, and the recommended remediation steps for their correction.

The main steps of web application vulnerability testing:

- **Information gathering/OSINT:** it consists of running specific searches, extracting and analyzing metadata, identifying IP address or IP address ranges, passively discovering running services and their version, domain information discovery and certification information analysis.
- **Discovery**: during the discovery phase, the running services are explored and analyzed by active scanners. The framework/application is identified and then the known vulnerabilities are detected by running automatic vulnerability scanning engines. The vulnerabilities are always manually validated by cybersecurity experts.
- **Configuration testing:** the main aim of the configuration testing is to determine the components of the framework/application and to test them for known vulnerabilities and verify the correctness of the configuration settings. It is also necessary to test the access for administrative pages and examine the presence of old and backup files in the file system.
- **Authentication testing**: the purpose of this phase is to determine whether the authentication scheme can be bypassed by an attacker and whether the lockout mechanism is properly working, or is it possible retrieve and reuse the users' or administrators' login information. The analysis of the session tokens and cookies is also the scope of this testing phase.
- **Input validation testing**: during the testing, various user inputs are handed to the application. By these requests, the input validation mechanism of the application is tested. The aim of the test is to verify whether the application allows the attacker to bypass the input validation mechanism (if it presents) and launch different forms of attacks (Cross Site Scripting, SQL Injection, XML Injection, Command Injection, Code Injection, File Inclusion).
- **Error handling:** during this phase, the experts are testing the error handling of the application. They validate the presence of error handling and analyze the content of error messages, whether in case of an unexpected error, the response of the application is correctly handled and the response does not contain sensitive information.

- **Cryptography testing:** the purpose of the testing is to find channels where data is sent unencrypted and to identify whether cryptographically strong algorithms are forced to be used. It also determines whether the used algorithms are supported by NIST or whether they have known vulnerabilities in their protocol, which is usually exploited by attackers to mount further attacks.
- **Application logic testing:** The aim of the testing is to identify the logic errors of the infrastructure/application and provide information, how can they be exploited to gain sensitive information.

Tools and utilities used for the testing

- Automatic vulnerability scanners (pl. Acunetix, Nexpose)
- Port scanners (pl. nmap, masscan)
- Network sniffing tools (pl. Wireshark, BurpSuite Professional)
- OSINT tools (pl. Shodan, Spiderfoot)
- Proprietary scripts and applications.