Network Working Group
Internet-Draft
Expires: April 20, 2003

A. Barbir
Nortel Networks
O. Batuner
Independent consultant
T. Chan
B. Srinivas
Nokia
H. Orman
Purple Streak Development

October 20, 2002

Security Threats and Risks for Open Pluggable Edge Services
draft-ietf-opes-threats-00

Status of this Memo

Copyright Notice

Abstract

The document investigates the security threats associated with OPES. The effects of security threats on the underlying architecture are discussed. The document does not specify or recommend any solutions. Proposed solutions are viewed as illustrations of the nature of threats.

Table of Contents

1. Introduction

The Open Pluggable Edge Services (OPES) [1] architecture enables cooperative application services (OPES services) between a data provider, a data consumer, and zero or more OPES processors. The application services under consideration analyze and possibly transform application-level messages exchanged between the data provider and the data consumer. The OPES processor can distribute the responsibility of service execution by communicating and collaborating with one or more remote callout servers. The details of the OPES architecture can be found in [1].

Security threats with respect to OPES can be viewed from different angles. There are security risks that affect content consumer applications, and those that affect the data provider applications. These threats affect the quality and integrity of data that the applications either produce or consume. On the other hand, the security risks can also be categorized into trust within the system (i.e. OPES service providers) and protection of the system from threats imposed by outsiders such as hackers and attackers. Insiders are those parties that are part of the OPES system. Outsiders are those entities that are not participating in the OPES system.

It is up to OPES service providers to verify the trust relationship between them, whereby intentional false information is tracked. Insiders can intentionally or unintentionally inflect harm and damage on the data consumer and data provider applications. This can be through bad system configuration, execution of bad software or, if their networks are compromised, by hackers.

Depending on the deployment scenario, the trust within the OPES system is based on a transitive trust between the data provider application, the OPES entities and the data consumer application. Threats to OPES entities can be at the OPES flow level and/or at the network level.

In considering threats to the OPES system, the document will follow a threat analysis model that identifies the threats from the prospective of how it will affect the data consumer and the data provider applications.

The main goal of this document is threat discovery and analysis. The document does not specify or recommend any solutions. Proposed solutions are viewed as illustrations of the nature of threats.

It is important to mention that the OPES architecture has many similarities with other so called overlay networks, specifically web caches and content delivery networks (CDN) (see [2] , [4]). This

document focuses on threats that are introduced by the existence of the OPES processor and callout servers. Security threats specific to content services that do not use the OPES architecture are considered out-of-scope of this document. However, this document can be used as input when considering security implications for web caches and CDNs.

The document is organized as follows: Section 2 discusses threats to OPES data flow on network and application level, section 3 discusses threats to other parts of the system and section 4 discusses problems related to the cryptographic data protection paying special attention to the hop-by-hop versus end-to-end protection.

2. OPES Data Flow Threats

Threats to OPES data flow can affect the data consumer and data provider applications. At the OPES flow level, threats can occur at Policy Enforcement Points and Policy Decision Points [3] and along the OPES flow path where networks elements are used to process the data.

A serious problem is posed by the very fact that the OPES architecture is based on widely adopted protocols (HTTP is used as an example). The architecture document specifically requires that "the presence of an OPES processor in the data request/response flow SHALL NOT interfere with the operations of non-OPES aware clients and servers". This greatly facilitates OPES deployment but on the other hand a vast majority of clients (browsers) will not be able to exploit any safeguards added as a base protocol extensions.

In reality active investigative actions are not feasible for the regular end user (Where this content comes from? Can I get it another way? What is the difference? Is it legitimate?). Even if there are facilities and technical expertise present to pursue these questions such thorough examination of each result is prohibitively expensive in terms of time and effort. OPES-aware content providers may try to protect themselves by adding verification scripts and special page structures. OPES-aware end users may use special tools. In all other cases (non-OPES aware clients and servers) protection will rely on monitoring services and investigation of occasionally discovered accidents.

OPES system poses a special danger as a possible base for classical man-in-the-middle attack. One of the reasons why such attacks are relatively rare is difficulty in finding an appropriate base: a combination of a traffic interception point controlling consistent flow of data and an application codebase with sufficient performance to analyze and possible modify all passing data. OPES processor perfectly meets this definition. This calls for a special attention to protection measures at all levels of the system.

Any break into an OPES processor or remote callout server can have a ripple effect on the integrity of the affected OPES services across all service providers that use the service. To mitigate this threat appropriate security procedures and tools (e.g., a firewall) should be applied.

Specific threats can be at the network level and at OPES data flow level.

2.1 OPES Flow Network Level Threats

OPES processor and callout servers are susceptible to network level attacks from outsiders or from the networks of other OPES service providers (i.e. if the network of a contracted OPES service is compromised).

OPES architecture is based on common application protocols that do not provide strong guarantees of privacy, authentication, or integrity. The IAB considerations [4] require that the IP address of an OPES processor be accessible to data consumer applications at the IP level. This exposes the OPES processor including remote callout servers to network level attacks. Use of TCP/IP as network level protocol makes OPES processors subject to many known attacks, like IP spoofing and session stealing.

The OPES system is also susceptible to a number of security threats that are commonly associated with network infrastructure. These threats include snooping, denial of service, sabotage, vandalism, industrial espionage, theft of service and inadequate system configuration that leaves unneeded ports and services open to the public.

There are best practice solutions to mitigate network level threats. It is recommended that the security of the OPES entities at the network level be enhanced using known techniques and methods that minimize the risks of IP spoofing, snooping, denial of service and session stealing.

In the following subsections we take a more detailed look at these threats and potential resulting harm.

2.1.1 OPES device spoofing

A malicious node could send false information about itself masquerading as an OPES device. Alternatively, despite the presence of a genuine OPES device which has been authenticated, the actual data transformation could be performed in a malicious colocated callout server which is resident in the same administrative domain as the OPES device. Furthermore, the malicious node could force the consumer or producer to use the services of a malicious OPES processor, which might render undesired or very expensive transformation services.

As a consequence, the malicious device would be able to eavesdrop on all traffic between the end-systems. In addition, unexpected and undesirable data transformation by the malicious processor or callout server would result. Finally, the malicious entity, that

successfully spoofs an OPES processor (or callout server), may refuse to forward the legitimate traffic to the content consumers, resulting in a Denial-of-Service attack.

2.1.2 Remote callout server spoofing

Similar to the threat described in 2.1.1, a malicious node could masquerade as a remote callout server. Despite the presence of an authenticated OPES device, the malicious data transformation could be performed in a remote callout server.

The effect of having such a malicious remote callout server is very similar to those produced by having a malicious OPES device or colocated callout server (see 2.1.1).

2.1.3 Session Hijacking

If a TCP/IP session is hijacked by an attacker, it would be possible for the hijacker to compromise the integrity of content on an OPES processor.

2.1.4 Threats to data confidentiality (eavesdropping)

An eavesdropper is typically capable of snooping on fields within messages in transit. Using various eavesdropping techniques, he may be able to garner various kinds of information including topology/ location/IP addresses etc. that may not be desirable to divulge. He also may be able to eavesdrop on the content messages being delivered to the consumer. Furthermore, to ensure secure data traversal from the provider to the consumer, authentication information must be exchanged between the provider and the consumer. When such security related information has to traverse through an OPES system, it is also subject to the threat of being eavesdropped on by the malicious entity.

2.1.5 Denial-of-Service (DoS)

The processor or the callout server can be overloaded by spurious service requests issued by a malicious node, which denies the legal data traffic the necessary resources to render service. The resources include CPU cycles, memory, network interfaces, etc. A Denial-of-Service attack can be selective, generic or random in terms of which communication streams are affected.

Distributed DoS is also possible when an attacker successfully directs multiple nodes over the network to initiate spurious service requests to an OPES processor (or call-out server) simultaneously.

2.1.6 Threats to network robustness

if OPES implementation does violate end-to-end addressing principles, it could endangers the Internet infrastructure by complicating routing and connection management. If it does not use flow-control principles for managing connections, or if it interferes with end-to-end flow control of connections that it did not originate, then it could causing Internet congestion.

An implementation that violates IAB requirement of explicit IP level addressing (for example by adding OPES functional capabilities to an interception proxy) may defeat many protective mechanisms and safeguards built into the OPES architecture.

2.2 OPES Flow Application Level Threats

At the content level threats to the OPES system can come from outsiders or insiders. The threat from outsiders is always intentional. Threats from insiders can be intentional or due to inappropriate implementations such as programming and configuration errors that result in bad system behavior.

Application level problems and threats to the OPES systems are discussed below:

2.2.1 Unauthorized OPES entities

Although one party authorization is mandated by the OPES architecture such authorization occurs out-of-band. Discovering the presence of an OPES entity and verifying authorization requires special actions and may present a problem.

Adding notification and authorization information to the data messages (by using base protocol extensions) may help, especially if the UserAgent software is aware of such extensions.

2.2.2 Unauthorized actions of legitimate OPES entities

According to the OPES architecture the authorization is not tightly coupled with specific rules and procedures triggered by the rules. Even if a requirement to approve each particular rule and procedure was set it looks at least impractical if not impossible to request such a permission from the end user. The authorization is given essentially for the class of transformations. The actual rules and triggered procedures may (maliciously or due to a programming error) perform actions that they are not authorized for.

2.2.3 Unwanted content transformations

An authorized OPES service may perform actions that do not adhere to the expectations of the party that gave the authorization for the service. Examples may include ad flooding by a local ad insertion service or use of inappropriate policy by a content filtering service.

On the other hand an OPES entity acting on behalf of one party may perform transformations that another party deems inappropriate. Examples may include replacing ads initially inserted by the content provider or applying filtering transformations that change the meaning of the text.

2.2.4 Corrupted content

The OPES system may deliver outdated or otherwise distorted information due to programming problems or as a result of malicious attacks. For example, a compromised server, instead of performing OPES service, may inject a bogus content. Such actions may be an act of cyber-vandalism (including virus injection) or intentional distribution of misleading information (such as manipulations with financial data).

A compromised OPES server or malicious entity in the data flow may introduce changes specifically intended to cause improper actions in the OPES server or callout server. These changes may be in the message body, headers or both. This type of threat is discussed in more detail below.

2.2.5 Threats to message structure integrity

An OPES server may add, remove or delete certain headers in a request and/or response message (for example to implement additional privacy protection or assist in content filtering). Such changes may violate end-to-end integrity requirements or defeat services that use information provided in such headers (for example some local filtering services or reference-based services).

2.2.6 Granularity of protection

OPES services have implicit permission to modify content. However, the permissions generally apply only to portions of the content, for example, URL's between particular HTML tags, or text in headlines, or URL's matching particular patterns. In order to express such policies, one must be able to refer to portions of messages and to detect modifications to message parts.

Because there is currently very little support for policies that are expressed in terms of message parts, it will be difficult to attribute any particular modification to a particular OPES processor, or to automatically detect policy violations.

A fine-grained policy language should be devised, and it could be enforced using digital signatures. This would avoid the problems inherent in hop-by-hop data integrity measures.

2.2.7 Risks of hop-by-hop protection

OPES services cannot be applied to data protected with end-to-end encryption methods because, by definition, the decryption key cannot be shared with opes processors. This means that if the endpoint policies permit OPES services, the data must either be transmitted without confidentiality protections or else with an alternative to end-to-end encryption: hop-by-hop encryption. In the latter case, all the parties in the OPES processing path must understand the encryption requirement and negotiate encrypted connections with their OPES partners.

Hop-by-hop protection is less effective than end-to-end protection, because any processor in the path can violate the confidentiality or integrity of the data without detection.

If a pair of processors in the delivery path use weak cryptography or manage keys poorly, there is a danger of data leakage. For this reason, different cryptographic keys should be used for each leg of the data stream.

Even if the data is not confidential, one might desire some checks on data integrity, to avoid modifications by unauthorized parties. The comments above apply to the use of end-to-end integrity, if it is based on shared-key cryptography. Again, it should be possible to use hop-by-hop data integrity to protect data as it moves between protection domains.

Currently there is no method to signal hop-by-hop encryption requirements. Either this must be added to the application protocol, or OPES must define its own signaling protocol, or all OPES traffic MUST ALWAYS be encrypted.

2.2.8 Threats to integrity of complex data

The OPES system may violate data integrity by applying inconsistent transformations to interrelated data objects or references within the data object. Problems may range from a broken reference structure (modified/missing targets, references to wrong locations or missing

documents) to deliberate replacement/deletion/insertion of links that violate intentions of the content provider.

2.2.9 Denial of Service (DoS)

The data consumer application may not be able to access data if the OPES system fails for any reason.

A malicious or malfunctioning node may be able to block all traffic. The data traffic destined for the OPES processor (or callout server) may not be able to use the services of the OPES device. The DoS may be achieved by preventing the data traffic from reaching the processor or the callout server.

2.2.10 Tracing and notification information

Inadequate or vulnerable implementation of the tracing and notification mechanisms may defeat safeguards built into the OPES architecture.

Tracing and notification facilities may become a target of malicious attack. Such an attack may create problems in discovering and stopping other attacks.

The absence of a standard for tracing and notification information may present an additional problem. This information is produced and consumed by the independent entities (OPES servers/user agents/ content provider facilities). This calls for a set of standards related to each base protocol in use.

3. Threats to out-of-band data

The OPES architecture separates a data flow from a control information flow (loading rulesets, trust establishment, tracing, policy propagation, etc.). There are certain requirements set but no specific mechanism is prescribed. This gives more flexibility for implementations but creates more burden for implementors and potential customers to ensure that specific implementation meets all requirements for data security, entity authentication and action authorization.

In addition to performing correct actions on the OPES data flow any OPES implementation has to provide an adequate mechanism to satisfy requirements for out-of-band data and signaling information integrity.

Whatever the specific mechanism may be, it inevitably becomes subject to multiple security threats and possible attacks. The way the threats and attacks may be realized depends on implementation specifics but the resulting harm generally falls into two categories: threats to OPES data flow and threats to data integrity.

The specific threats are:

3.1 Threats that endanger OPES data flow

Any weakness in security, authentication and authorization mechanism implementation may open a possibility to threats and attacks described in section 2.

An OPES system implementation should address all these threats and prove its robustness and ability to withstand malicious attacks or networking and programming problems.

3.2 Inaccurate Accounting Information

Collecting and reporting accurate accounting data may be vital when OPES servers are used to extend a business model of content provider, service provider or as a basis for third party service. Ability to collect and process accounting data is an important part of OPES system functionality. This functionality may be challenged by distortion or destruction of base accounting data (usually logs), processed accounting data, accounting parameters and reporting configuration.

As a result a data consumer may be inappropriately charged for viewing content that was not successfully delivered, or a content provider or independent OPES services provider may not be compensated

for the services performed.

OPES system may use accounting information to distribute resources between different consumers or limit resource usage by a specific consumer. In this case an attack on accounting system (by distortion of data or issuing false configuration commands) may result in incorrect resource management and DoS by artificial resource starvation.

3.3 OPES service request repudiation

An entity (producer or consumer) that is authorized to make a certain request to the OPES processor claims, later, that it did not make that request. As a result an OPES entity may be held liable for unauthorized changes to the data flow.

3.4 Exposure of private information

The OPES system may inadvertently or maliciously expose private information such as (passwords, buying patterns, page views, and credit card numbers) of the data consumer. Logs and accounting data may also contain sensitive private information.

3.5 Inconsistent privacy policy

The OPES entities may have privacy policy not consistent with end user or content provider expectations.

Privacy related problems may be further complicated if OPES entities, content providers and end users belong to different jurisdictions with different requirements and different levels of legal protection. As a result the end user may not be aware that he/she does not have the expected legal protection. The content provider may be exposed to legal risks due to a failure to comply with regulation which he is not even aware of.

3.6 Exposure of privacy preferences

The OPES system may inadvertently or maliciously expose end user privacy settings and requirements.

3.7 Exposure of security settings

There are risks that the OPES system may expose end user security settings when handling the request and responses.

3.8 Improper enforcement of privacy and security policy

OPES entities are part of the content distribution system and as such take on certain obligations to support security and privacy policies mandated by content producer and/or end user. However there is a danger that these policies are not properly implemented and enforced. The data consumer application may not be aware that its protections are no longer in effect.

Privacy and security related parts of the systems are most often targeted by malicious attacks and ability to withstand such attacks is of paramount importance.

4. Security Considerations

This document discusses multiple security and privacy issues related to the OPES services.

References

[1] A. Barbir et. al, "An Architecture for Open Pluggable Edge Services (OPES)", Internet-Draft: http://www.ietf.org/internet-drafts/draft-ietf-opes-architecture-03.txt, Aug 2002.

[2] A. Barbir et. al, "OPES Use Cases and Deployment Scenarios", Internet-Draft: http://www.ietf.org/internet-drafts/draft-ietf-opes-architecture-02.txt, July 2002.

[3] A. Barbir et. al, "Requirements for Policy, Authorization and Enforcement of OPES Services", Internet-Draft: http://www.ietf.org/internet- drafts/draft-ietf-opes-authorization-00.txt , October 2002.

[4] Floyd, S. and L. Daigle, "IAB Architectural and Policy Considerations for Open Pluggable Edge Services", RFC 3238, January 2002.

[5] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[6] Cooper, I., "Internet Web Replication and Caching Taxonomy", RFC 3040, January 2001.

Authors' Addresses

Abbie Barbir
Nortel Networks
3500 Carling Avenue
Nepean, Ontario K2H 8E9
Canada

Phone: +1 613 763 5229
EMail: abbieb@nortelnetworks.com

Oskar Batuner
Independent consultant

EMail: batuner@attbi.com

Tat Chan
Nokia
5 Wayside Road
Burlington, MA 01803
USA

EMail: Tat.Chan@nokia.com

Bindignavile Srinivas
Nokia
5 Wayside Road
Burlington, MA 01803
USA

EMail: bindignavile.srinivas@nokia.com

Hilarie Orman
Purple Streak Development

Phone:
EMail: ho@alum.mit.edu

Appendix A. Acknowledgements

The authors gratefully acknowledge the contributions of: Markus Hofmann

TBD

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.