

ODON: An On-Demand Security Overlay for Mission-Critical Applications

Jinu Kurian*, Ajay Kulkarni*, Hai Trong Vu*, and Kamil Sarac†

*Dept. of Computer Science, University of Texas at Dallas, Richardson, Texas- 75080

Email: {jinuk, hai.vu, ajay.kulkarni}@student.utdallas.edu

† Email: ksarac@utdallas.edu

Abstract—In this paper we consider the construction of a large-scale, highly available and secure overlay network to enable mission-critical communication between emergency personnel at a disaster area and their coordinating agencies across the Internet. This network is designed to be secure against network-based failures and external attacks including denial of service (DoS) attacks. We design several protocols for the effective operation of the network, including protocols for user access verification and the establishment of session credentials between the user and the target server. We verify these protocols theoretically for their security and evaluate the overall performance of the system with a combination of simulation and implementation.

I. INTRODUCTION

In the event of an emergency situation or disaster event such as floods, earthquakes, hurricanes or terrorist attacks, it becomes imperative to expedite recovery operations to the affected sites in a timely manner. Therefore, it becomes vital that an effective communication medium is prevalent between various distributed disaster management organizations and emergency response personnel at the affected sites [1]. It is equally vital to consider the resiliency of this communication medium to interference in the form of attacks on the medium itself or general failures in the medium itself like outages and congestion.

The requirements for establishing an emergency telecommunication service (ETS) have been explored by the IETF Internet Emergency Preparedness (IEPREP) society [1]. Despite the incentives for the establishment of a comparable Internet emergency service [2], there has been little rigorous work in the area [3]. Several factors can be cited for this discrepancy between ETS and a comparable Internet emergency service system. These include non-cooperation from ISPs for the establishment of a separate network, higher reliability and security concerns in the regular Internet compared to PSTN and the fiscal and resource wastage issues related to deploying a network which will be used only during emergency situations. Overlay networks offer a feasible solution to some of these issues. As we discuss in the remainder of the paper, overlay networks can be deployed with minimal participation from the ISPs, offer services like reliability, QoS guarantees and DoS resistance and can be activated in an on-demand manner to prevent wastage of resources.

In this paper we consider the problem of creating an on-demand overlay network for mission-critical communication requirements. Paramount in our design considerations is the establishment of a network which is resilient to failures in the network and cyberterrorism in the form of DoS and break-in attacks on the entities of the system. Our primary focus is to efficiently navigate the strict security and reliability requirements

of such an emergency network while respecting the limitations introduced by the on-demand nature of the application. We concentrate primarily on the communication between a set of fixed sites (e.g. various governmental agencies) and a small set of pre-authorized users over the wired portion of the Internet. Additional design considerations include effective management of nodes in the overlay network, support for mobility of users, protecting the target site(s) from unicast based flooding attacks and minimizing collateral damage due to compromised nodes and users.

The rest of the paper is organized as follows. In Section II we look at related work in mission critical networks. In Section III we discuss the requirements and restrictions that we aim to meet in our design. In Section IV we introduce the ODON overlay architecture, its management, operation and design rationale. In Section V we look at the various functional components of ODON. In Section VI we analyze the security properties of the system. Section VII evaluates the performance of the system under various conditions through a combination of simulations and implementation. Finally, Section VIII concludes the paper.

II. RELATED WORK

In general, related work in mission-critical networks can be categorized into two: 1) mechanisms for the last-hop which deal directly with the traffic in the affected area and 2) end-to-end (e2e) mechanisms from the affected site to locations in the wired Internet.

There has been a significant amount of work in the area of wireless ad-hoc and sensor networks which deal with effectively reestablishing broken communications in last-hop. Since ad-hoc networks are beyond the scope of our paper, interested readers are referred to [4] for details. In e2e solutions, there has been very little work to deal with the nuances of the mission-critical application scenario. The first work which suggested the possibility of using overlay networks in mission-critical applications was by Gao and Beard in [5]. Other comparable work exists primarily in resilient routing and DoS resistant communication scenarios which have inspired us in our solution. We discuss some of the important ones below.

The use of overlay networks as a mechanism to improve resiliency in the Internet was first proposed in RON [6]. RON proposed a distributed overlay network to route around network failures and find better paths for higher resiliency than unicast based communication. Single-hop source routing [7] has demonstrated that the benefits of a routing overlay similar to RON can be obtained with a single level of indirection at a much lower cost.

SOS [8] was the seminal work in the use of overlay networks for DoS resistant communication. It utilizes an overlay network with circuitous routing to hide the path, a filter ring around the protected target and a set of secret nodes which are allowed to pass through the filter ring. SOS, while effective and secure has high overhead due to its routing, high redundancy if used for multiple destinations and does not deal effectively with the possibility of compromised overlay nodes.

OverDoSe [9] is another overlay based system for DoS defense. In OverDoSe overlay nodes and clients are required to execute puzzles which aim to prevent any particular node from flooding the destination. The protected target in OverDoSe is required to be isolated in a non-unicast environment with only overlay nodes being able to access the target through the filter. A client chooses an overlay node which directs the user traffic through the filter towards the destination. As we will see later in Section VII, OverDoSe is vulnerable to large scale distributed attacks during its connection establishment phase. ODon performs at par or better than OverDoSe in all other scenarios and offers simpler solutions to compromised nodes.

III. DESIGN REQUIREMENTS AND RESTRICTIONS

In this section we describe the general requirements we consider in our design and some of the restrictions necessitated by the application context. Some of the requirements are suggested in RFC 3689 [1] as part of the requirements for an emergency telecommunication service (ETS) framework proposed by the IEPREP IETF charter. We also define the terminology that will be used in the remainder of the paper.

In the rest of our discussion, 'system' refers to the ODon overlay network, 'overlay nodes' refers to the deployed machines which create the ODon overlay network, 'user or authorized user' refers to an authorized user of the system who once verified becomes a 'verified user', 'destination' refers to the target node, i.e. the government agency site the authorized user desires to communicate with, 'normal operating circumstances' refers to a non-emergency situation during which the system is not required for communication, 'emergency scenario' refers to a disaster event or other mission-critical application scenario which necessitates the use and presence of the emergency network.

A. Attacker threat model and operational assumptions

We assume that the last-hop at the affected area has (possibly through an ad-hoc network) established communication to an access point in the wired Internet. We assume that there is a filter ring consisting of high-speed programmable routers at the destination domain which can be configured to filter out undesired traffic with ACLs or null routing. We also assume that the destination server has provided a set of users (during normal operating circumstances) with access control credentials (see Section V-A for details) which can be used by the overlay nodes to verify the user's authorization.

The attacker is assumed to have the ability to eavesdrop, record, replay or in other ways subvert the traffic in the system. It can also pretend to be any of the legitimate entities in the system and may be able to execute all or some of the steps

in the access verification and token exchange protocols. The attacker may have knowledge of the location of every node in the ODon system and the target being protected. The attacker may be able to obtain legitimate credentials for the system. The attacker may have the ability to break-in and compromise any entity in the system except the protected destination. However it is assumed that the number of un-compromised nodes is at least twice the number of compromised nodes.

The attacker may have the ability to bring down through flooding attacks no more than one-third the number of nodes including the nodes currently in use in the system at any given time. The attacker can flood the destination server with unwanted traffic in an attempt to bring down the server. The attacker can also launch directed attacks against nodes being currently used in the system instead of randomly chosen nodes. We assume that the attacker can in worst case time $O(n) * \text{average rtt}$, correctly identify and specifically target an in-use overlay node. Here n is the number of overlay nodes in the system and rtt is the round trip time of a probe packet. The attacker does not have the ability to break into the destination server or its Key Distribution Center (KDC). It also does not have the ability to bring down the core of the Internet or the access point (there can be several) at the last-hop to the wired Internet.

B. Design requirements

High availability is one of the most crucial requirements of mission-critical/disaster management systems [10]. In the Internet, this translates to a high end-to-end packet-level availability which is defined as the ratio of the total number of packets sent to the number of packets received at the destination. The system should be highly available even during aberrant circumstances like link failures, high congestion and DoS attacks on the system.

In general, the system needs to be secure against an attacker with capabilities as described in Section III-A. Security of the system therefore has to be three fold: 1) The system should be used only by authorized users of the system, 2) authorized but malicious users and/or compromised overlay nodes cannot break the system and 3) the system and the destination site must be resilient to DoS attacks.

In addition to these desired requirements, the application scenario also introduces various restrictions which introduce additional requirements. Fixed systems, i.e. a system with permanently live nodes, long lived secure connections, bandwidth guarantees or MPLS tunnels cannot be used. The system should be essentially dormant under normal operating circumstances and be live only when activated (on-demand) during a critical event. This is necessary because the system is to be utilized only during the emergency scenario and should not waste resources for its existence under normal circumstances.

High end-to-end latency for communication should also be avoided. Though latency and user friendliness are not primary requirements, high latency limits the effectiveness and type of services available to efficiently coordinate between the user and the management agencies (e.g. a video conference between various departmental figureheads and the user).

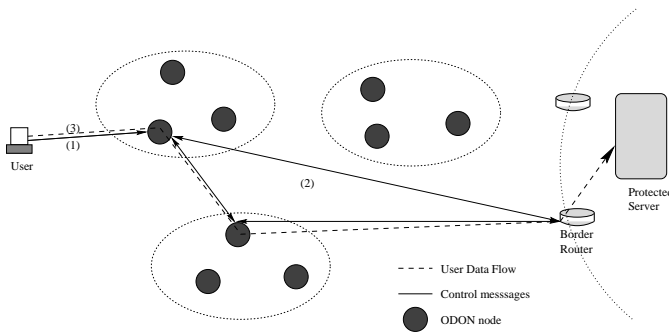


Fig. 1. Overlay topology: In the figure (1) is the access verification step, (2) is the session token exchange step, (3) is the flow of user data

IV. ODON: AN ON-DEMAND OVERLAY

In this section we introduce the ODON overlay. We present the ODON architecture and an overview of its operation.

A. Architecture

Communication over ODON involves four entities: 1) users who are pre-authorized by the destination server and require effective communication with the destination for disaster recovery management, 2) overlay nodes deployed across the various domains in the Internet, 3) the destination server and 4) an ODON client installed at the user's machine (see Figure 1).

Unlike typical overlay networks, the overlay nodes do not form a pre-determined overlay network and do not forward user data across the overlay network. The overlay level connectivity in ODON is established on-demand and used only to exchange signalling and other management information between the overlay nodes. Based on the operational requirement, every node in ODON can act as either: 1) *access* node or 2) *forwarding* node. When acting as an *access* node, the overlay node serves as the authentication and access control entity for authorized users in the system. When acting as a *forwarding* node, the overlay node acts as a proxy. It verifies the user's session token and forwards the user provided data through the filter ring to the destination in a one-hop source routing manner.

On-hop source routing has been shown to give comparable performance results to multi-hop overlays in providing higher resiliency and response times to unicast based communication [6], [7]. In ODON we choose to use the one-hop source routing approach because of its high performance and intrinsically simple operation. This is important because the on-demand nature of the system precludes complex routing schemes and established tunnels between overlay nodes.

The ODON client is implemented as a proxy application in the user's machine and at the destination server. All traffic going in and out of the system is intercepted by the ODON client and manipulated to work with the ODON overlay.

The node organization in ODON is flat but separated into domains based on node proximity, i.e. node addressing is organized in the form "domain.endsystem" with nodes in a domain being aware of each other. This addressing scheme allows us to quickly search or disseminate information to all

the nodes in a domain, for e.g. the partial keys during the access verification procedure (Section V-A).

B. Operation

From a high-level perspective, the operation of ODON is as follows. In the event of an emergency scenario, an authorized user or a destination node chooses one of the overlay nodes to act as its *access* node. The authorized user's credentials are verified (Section V-A) by the *access* node which then executes the token exchange protocol (Section V-B). After the user's credentials are verified, the *access* node chooses a set of overlay nodes to act as *forwarding* nodes for the user and executes a three-way token exchange protocol between the user, the *forwarding* node and a border router at the destination domain. The objective of this token exchange protocol is to establish a session token and a session key to be used for the rest of the session between the user and the destination. At the completion of the token exchange protocol, the user is provided with a session token that is forwarded along with the user data and verified by the *forwarding* nodes before data forwarding. The user and the destination are also provided with a session key that is used to provide integrity and confidentiality to user traffic.

The ODON client at the user end then replicates and forwards user traffic along with the session token to the chosen *forwarding* nodes. The *forwarding* nodes verify the user's session token and forward the user's traffic through the filter ring towards the destination. During the course of the session, a chosen *forwarding* node initiates another token exchange session between the user and another distinct set of overlay nodes. This allows the ODON client to rapidly switch from a poorly performing *forwarding* node to a new *forwarding* node when required.

The freedom to switch between any pair of overlay nodes is a unique and vital component of the ODON architecture. It allows the user to avoid poorly performing nodes which may be congested or subject to DoS attacks. It also helps in dealing with possibly compromised forwarding nodes as we will see in Section VI. To decide when to switch to a new *forwarding* node, every ODON client measures two metrics: i) RTT between the user and the destination per *forwarding* node and ii) decay, the total time a particular *forwarding* node has been in use during a given session. Based on these measured metrics, the ODON client maintains two additional metrics: i) Estimated RTT measured as $estrtt = 0.9 * estrtt + 0.1 * rtt$ and ii) Average decay. The ODON client switches to a new *forwarding* node if the measured $RTT = \alpha * estrtt$ or the decay is greater than the average decay, where $0 \leq \alpha \leq 1$ is the scaling factor.

V. FUNCTIONAL COMPONENTS

In this section we discuss in detail the components that make up the operation of the system. We discuss the protocols used by ODON for user access verification and session token exchange and traffic forwarding by the *forwarding* node after the session is established.

A. Access verification

The access verification step consists of the *access* node retrieving an authentication token installed in the user's machine and verifying it based on user entered information. This authentication token is a single-use token installed in the user's machine or provided in a storage device to the user by the KDC at the destination side during normal operating circumstances.

The access verification procedure at the *access* node effectively pushes a distributed filter deep into the network away from the destination. It ensures that the connection establishment process by itself cannot be used by an attacker for DoS attacks. It also helps protect the destination from directed flooding based DoS attacks. Finally it establishes a shared key between the user and the destination ensuring that only authorized users can utilize the ODon overlay and that the end-to-end communication is secure and confidential.

Key Distribution: During normal operating circumstances a KDC at the destination site generates a set of encrypted partial keys kp_i for distribution to the overlay nodes. The keys, like node addresses are indexed by domain with a range of key values being assigned to nodes based on their domain. This allows the access node to easily search for the partial keys by contacting any node within the desired domain. The partial keys are used to generate a set of encryption keys K_i . In general a key $K_i = kp_a \text{ xor } kp_b \text{ xor } \dots \text{ xor } kp_k$ such that no two partial keys belong to the same node. The rationale behind distributing the keys in this manner is to avoid the collateral damage that can occur if any of the overlay nodes are compromised albeit at a loss in performance. This performance hit can however be avoided at the cost of higher complexity and less scalability by using a (k,n) threshold key scheme [11] to distribute the keys. These encryption keys are used in the generation of a set of authentication tokens that are distributed to the authorized users. These authentication tokens will (see Section V-A) be used for the access verification at the *access* nodes.

Structure and security of authentication token: The structure of the installed token is shown in figure 2. The token has 8 fields, Domain, Exp(iry), Key index, Data, U(ser)name, P(ass)w(or)d, Seal and Sig(nature). The domain field contains the domain (i.e. the destination server) for which the token was issued and is valid for. Exp contains an explicit expiration time for the token in cleartext. Key index contains an index which references the partial keys used to generate the encryption key used in the creation of the token. Data field contains other optional information which can be used to specify additional information about the user's access privileges. Pwd and Uname are encrypted versions (using the encryption key) of the user's username and password used for access verification at the destination. These credentials are temporary and generated by the KDC only for use in the ODon system. Seal is a keyed-hash of the entire token keyed with a key derived from the encryption key and finally the Sig field is a signature created by the KDC over all the token fields except the Seal.

The authentication token described has some important properties that make it secure and non-transferable. The Do-

Domain	Key Info	Uname	Pwd	Exp	Data	Seal	Sig
--------	----------	-------	-----	-----	------	------	-----

Fig. 2. Structure of the Access Token

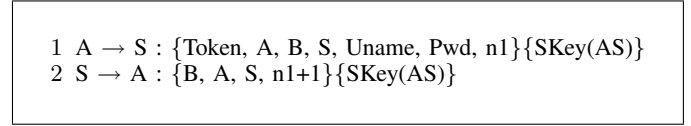


Fig. 3. Access verification protocol.

main field binds the token to a single domain. Even if the token is intercepted by an adversary, it is not useful without the username and password of the user. If required, the token can also be bound to a single user machine (for example using a combination of fields like BIOS checksum, MAC address, operating system version etc.) which ensures that it cannot be distributed to create an attack. The Seal field ensures that the token has not been altered in any manner by the user or an attacker. Since the seal includes the Expiry and Id fields as part of the hash, the user cannot modify the token to extend its validity or reuse an expired/revoked token. The Sig field created by the KDC ensures that the token cannot be manufactured by an adversary. Finally the partial keys used for creating the token are expired by the *access* node after the authentication process is completed by disseminating this information to the nodes which returned the partial keys during the search. This ensures that the token is invalid after its first use.

Authentication and access control: In ODon the authentication and access control procedure is done simultaneously when the user presents his token and provides his credentials.

The protocol is displayed in figure 3. In the figure, A refers to the user, S refers to the *access* node, SKey(AS) is a shared key between A and S derived from the password of user. After obtaining the token and decrypting the pwd field, the *access* node is also able to derive this key from the password.

As can be seen in this process, the user by providing its credentials and having them verified authenticates itself and establishes its identity. By virtue of it possessing a valid token, it additionally establishes its access privileges and thus completes the access verification step.

B. Token exchange protocol

The token exchange protocol is a three way protocol initiated by the *access* node after the user token is verified to establish a session token. This session token is shared by the user, its *forwarding* node and the border router at the destination. Additionally it also establishes a shared secret key and nonce between the user and the destination server.

The protocol is displayed in figure 4. In the figure, A is the user, B is the destination server, S is the *access* node, O is a *forwarding* node and R is the border router/KDC at the destination. In this process we assume that the *forwarding* node chosen are the nodes which returned the partial keys during the key search in the access verification procedure.

```

1 S → R : {S, R, A, B, O, n1}{SKey(SR)}
2 R → S : {R, S, A, B, n1+1, Token, {Ks, n3}{Skey(AR)}}
           {SKey(SR)}
3 S → O : {S, O, A, B, n2, Token}{SKey(SO)}
4 O → S : {O, S, A, B, n2+1, Token}{SKey(SO)}
5 S → A : {S, A, {Ks, n3}{Skey(AR)}, Token'}{Skey(AS)}

```

Fig. 4. Token exchange protocol.

Domain	Exp	ID	Seal1	Seal2
--------	-----	----	-------	-------

Fig. 5. Structure of the Session Token

This simplifies the key exchange process so that the shared key $SKey(SO)$ can be derived from the partial key returned. The shared key $SKey(SR)$ is assumed to be a long term shared key between the nodes in a domain and the border node and $Skey(AR)$ is a shared key between the user and the KDC created based on the user's credentials at the KDC.

The protocol proceeds as follows. The *access* node contacts the border router which returns the session token and a ticket consisting of a session key and nonce encrypted with the shared key between the user and the KDC. The *access* node then contacts the *forwarding* node(s) with the session token. The session token is modified (with the Seal2 field) by the *forwarding* node and returned to the *access* node. Finally the *access* node returns the token and the ticket to the user for its session.

The structure of the session token is shown in figure 5. Domain and Exp are similar to the access token. The ID field binds the token to a single *forwarding* node. The Seal2 field is created by the *forwarding* node over the data in the token while the Seal1 field is created by the border router. Since the token is valid only at a single *forwarding* node and unusable after the session is completed due to its short expiration time, replay and man-in-the-middle attacks are avoided here. The token exchange protocol is repeated with one of the active *forwarding* nodes acting as the initiator with another set of nodes for fast node switching.

The ODon client at the user receives the session token, a nonce and a session key at the end of the token exchange protocol. The ODon client encrypts the data provided by the user with the session key along with the nonce and sends it to the *forwarding* node along with the session token. The process is repeated if there are multiple *forwarding* nodes so that the data is replicated between all *forwarding* nodes.

A *forwarding* node upon receiving the user data initially verifies the session token. It does so by recomputing the Seal2 field and the ID field. If valid, it forwards the data towards the destination server through the border router. The router again verifies the token by recomputing the Seal2 field and if verified forwards the traffic to the destination server. The reverse side traffic also follows the same path through the *forwarding* node to the user. Note that since the session token is valid only at a single *forwarding* node, the token can be sent in cleartext

and is useless to an attacker.

VI. SECURITY ANALYSIS

In this section we use theoretical protocol modeling techniques to analyze the security of the access verification and token exchange protocol described in the previous section. Finally we discuss the security properties of the system and its effectiveness in dealing with compromised nodes.

A. Theoretical perspective

In this subsection we use formal verification mechanisms to evaluate the correctness of our two proposed protocols. Since the access verification protocol is trivial and our space is limited, in this section we concentrate on the three-way token-exchange protocol. Details about the verification of the access verification protocol can be found in [12]. Specifically, we model the interactions between the participating entities as communications between a set of sequential processes. This allows us to verify that the shared secrets (password, nonce, session key) are shared only between the participants of the protocol and not to an intruder.

The language of Communicating Sequential Processes (CSP) [13] can be used to describe any system with agents that communicate by passing messages between each other. Protocols specified in CSP are usually analyzed by the failures divergence refinement (FDR) [14] model checker for the analysis of security protocols [15], [16], [17].

Our analysis of the token exchange protocol proceeds as below. The first requirement is a set of specifications. These specifications are the properties that the protocol is required to maintain at the end of its run. In our protocol, the final result of the protocol run should be that S (the access node) has successfully established a token between O (the *forwarding* node), R (the remote KDC) and A (the user). Additionally, the protocol also needs to ensure that the session key Ks and the seed nonce n3 are both kept secret and shared only between A and R.

$Secret(R, n3, A)$

$Secret(A, n3, R)$

$Secret(A, Ks, R)$

$Secret(R, Ks, A)$

$Secret(A, Token, [O,S])$

$Secret(O, Token, [A,S])$

$Secret(S, Token, [A,O])$

$Agreement(A, R, [n3,Ks])$

$Secret(R, n3, A)$ specifies that R believes at protocol completion that the value of n3 is a secret shared between and A only. $Agreement(A, R, [n3, Ks])$ specifies that A and R have agreed on the values of n3 and Ks. These eight statements completely specify all the desired requirements of the protocol.

The next requirement is to model the intruder and the knowledge it possesses. From an intruder Mallory's perspective, she would know the participants in one of the protocol runs, i.e. it knows Alice the user, Ray the border router and Sam the *access* node. Additionally, she might also be an authorized user in the system, i.e. she possesses a shared secret key with the *access* node Sam. We assume that Alice's

secret and her shared key $S_{key}(Alice)$ are unknown to Mallory. Additionally, $S_{key}(Sam)$ is also unknown to Mallory.

$Intruder = Mallory$

$IntruderKnowledge = \{Alice, Ray, Mallory, Sam, S_{Key}(Mallory)\}$

The next requirement is to specify working systems to validate the protocol specification. These systems represent actual protocol runs between the different agents. It also specifies the knowledge that each agent possesses at the start of the protocol run. For example, a practical system is one in which the attacker can act both as initiator and responder in a single protocol run, or it can have multiple sessions concurrently or sequentially. The specification below shows an example of such a system:

$INITIATOR(Sam, Ray, Na)$

$INITIATOR(Sam, Ray, Na)$

$RESPONDER(Sam, Ray, Nb)$

$SERVER(Sam, Ks, Nb, Nc)$

In this case the attacker Mallory can initiate multiple sessions with the server Sam, or respond to a request from Alice masquerading as the server, masquerade as Sam and initiate sessions with Bob or pretend to be Bob and respond to messages from Sam. We consider six such systems with different combinations of agents acting as initiator or responder. Note that there are infinitely many such systems that can be created with different combinations of agents. However it is generally accepted [18] that these 6 systems will find nearly all attacks.

We evaluate the correctness of our protocol for each of the six systems and for all six specifications using FDR. All the six systems check correctly for all specifications suggesting that protocol implementation is secure against almost all known attacks in ensuring the secrecy of all desired values at the end of the protocol run.

B. Security properties of the system

In this subsection we discuss 5 important security properties possessed by the ODon system. These properties combine to make the system secure against many possible attacks.

Property 1: Dropping or delaying traffic by overlay nodes is circumvented by the ODon client. The ODon client at the user site monitors its chosen *forwarding* nodes for performance. If the *forwarding* node is dropping or delaying packets (either maliciously or due to network conditions), the ODon client forces a switch to another node.

Property 2: There are at least $2k+1$ *forwarding* nodes, where k is the maximum possible number of compromised nodes. Since each *forwarding* node is in use for a short period of time, if a node recovery mechanism is present on the suspicion of a compromised node, the assumption is also practically valid.

Property 3: Access tokens and session tokens are single use (Section V-A) and additionally session tokens are valid only at a single *forwarding* node. Since the key generating the token is expired after a successful authentication and session tokens are valid only for a very small period of time, they are both single use.

Property 4: The signature field in the access token can be verified by any overlay node. The signature is created by the destination KDC. It is assumed that all overlay nodes are provided with a valid certificate for the KDC.

Property 5: The border router(s) at the destination mediates all session requests to the target.

Attacks from compromised nodes and users: We now use the properties described above to discuss how the ODon system deals with compromised entities in the system.

Compromised users: A compromised user possesses a valid access token and user credentials. It can attempt to distribute the token to multiple attackers to flood the destination server with authenticated traffic through the overlay itself. However by property 3, the access token is single use and is expired after the first authentication. So only one of the attackers can ever be successfully authenticated with a valid access token. Forging the token is also implausible as we discussed earlier in section V-A. So, the damage that can be done by a compromised user is minimal.

Compromised access node: If the overlay node the user chooses as its *access* node is malicious, it can drop or delay the user's access request. But by property 1, the user can switch to another *access* node for its session request. A compromised node can coordinate with multiple distributed attackers to let in unauthorized traffic and flood the server. However by property 5, the border router at the destination mediates all session requests. So, traffic without the proper session token will be filtered out. Additionally the border node can restrict the number of token exchange sessions initiated by a particular *forwarding* node over a period of time. This additionally prevents the *access* node from obtaining a large number of valid session tokens to distribute to attackers. Finally as proven in Section VI-A, the session key and nonce are kept secret from the *access* node. So the session established between the user and the destination is kept secret even if the *access* node is compromised.

Compromised forwarding node: A compromised *forwarding* node can drop or delay the user's traffic to affect its performance. However by property 1, this attack is detected if it is significant enough to cause a performance loss for the user, and routed around. So the only viable option for a compromised *forwarding* node is to flood the destination. However by property 5, traffic without valid tokens will be dropped anyway and the *forwarding* node can be temporarily filtered out to prevent its further use. An alternative for the *forwarding* node would be to inject random or junk data into a valid data stream from the user to flood the destination. However by property 2 there are at least $2k$ uncompromised *forwarding* nodes in the session. So, the destination can monitor the amount of traffic sent over a period of time by the majority of overlay nodes. Since by property 1 packet dropping is not possible, any *forwarding* node injecting traffic into a valid stream can be detected based on the majority sending rate. A compromised *forwarding* node can also eavesdrop or record the traffic it forwards. However the traffic is encrypted using the session key which makes both implausible.

As can be seen in this section based on simple techniques

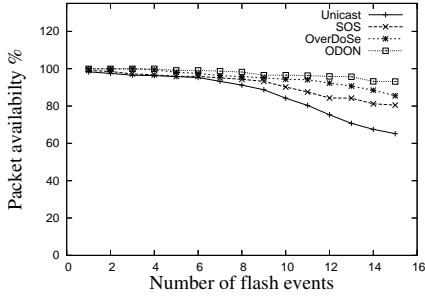


Fig. 6(a). Availability under emergency conditions.

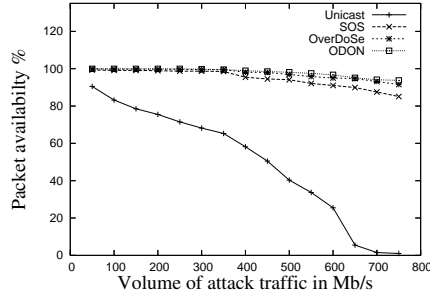


Fig. 6(b). Availability during DoS attacks.

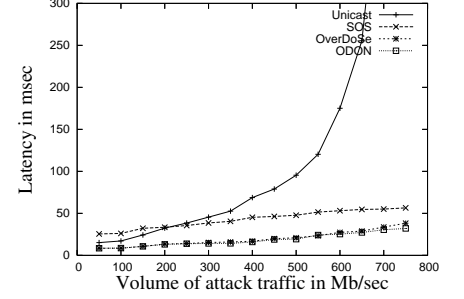


Fig. 6(c). Response time under DoS attacks.

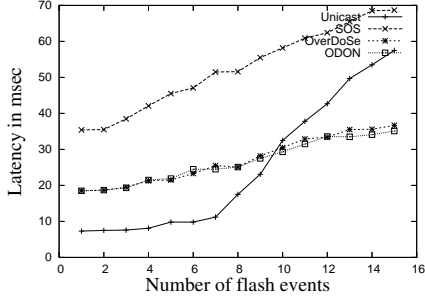


Fig. 6(d). Response time under emergency conditions.

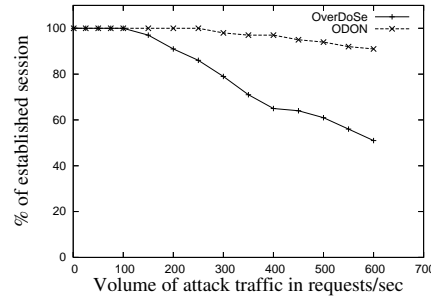


Fig. 6(e) Percentage of established sessions during DDoS attack.

and security properties instilled in the architecture, we are able to provide effective solutions for the avoidance of compromised nodes in the system. The freedom instilled to the user in the choice of access points and *forwarding* nodes allows us to provide a highly secure system.

VII. EVALUATIONS

A. Motivation

In this section we present simulation based analysis of the performance of the ODon system. In particular we aim to demonstrate that the requirements specified in Section III are met in our design of the ODon architecture. We measure the performance of the system across three specified requirements: 1) High Availability, 2) DoS resistance, 3) End-to-end latency introduced by the system.

B. Simulation setup

We use the BRITE topology generator to create a hierarchical top-down topology consisting of 10 AS's with 50 nodes in each AS for a total of 500 nodes. There are 10 users and 20 overlay nodes randomly distributed across all the domains in the topology. The destination server is randomly chosen from the remaining nodes.

The experiments are conducted under varying environmental conditions including: i) emergency operating environment which is characterized by flash crowds crossing the network and random failures ii) Flooding based DDoS attacks directed at the target server and overlay nodes and iii) DDoS attacks on the connection establishment process. All our simulations are repeated three times to account for the randomness and the results are averaged over all three runs.

C. Metrics

Packet level availability: Availability is defined as the ratio of the total number of packets sent to the number of packets received at the destination. This includes lost and retransmitted packets. For ODon since the traffic is replicated by two, if either or both of the replicated packets are received, the number of received packets is assumed to be two for two sent packets.

Response time: Response time is a measure of the time in msec for a request sent by the user to be responded to by the server.

D. Emergency environment

To model this scenario, in addition to the background traffic as in the normal case at a lower rate of 2 Mbps, we create flash events targeting various distributed servers in the topology [19]. We create this flash event by adding a new sender to a given destination every 0.01 seconds in the background traffic till it peaks, maintaining it for 2 seconds and ramping down till the sending rate is the base 2 Mbps. The flash events are repeated with more added senders with a 1 second gap in between each event to model larger and larger flash events.

E. DoS attacks

Denial of service attacks can target either the destination or the nodes in the system for the non-unicast cases. To create DoS attacks, the attacking senders send traffic at 10 Mbps from each sender to the destination and to the currently in-use overlay node for the non-unicast cases. In the unicast case the attack is concentrated on the target alone while in the other cases, the attack traffic directed at the target is dropped at

least 2 hops away from the final target. Based on our previous discussion in Section III-A, directed attacks on the *forwarding* node occur within a period exponentially distributed between 100 and 250 ms from the time the node becomes in use.

F. DDoS attacks on the connection establishment process

These experiments directly compare the connection establishment process of OverDoSe with ODon. In the case of OverDoSe, the attacker generates a large number of connection establishment requests from a large number of hosts such that the rate generated by each host is low enough to be not filtered out by the overlay node. This attack is replicated across multiple overlay nodes. Every request made by the attackers may either be filtered out by the overlay node if the total volume of traffic through the node is too high, or transmitted to the server which is then required to verify it and admit or filter out the request. The filtering out of requests by the overlay node has the effect of dropping legitimate requests also which use the same overlay node. This also increments the difficulty of the puzzles generated by the server throttling the sending rate of both attacker and legitimate user and additionally causes legitimate users to be preferentially discarded because the attackers solve equivalently high level of puzzles. In the case of ODon, all attackers are assumed to initially possess a legitimate token (which should never happen in a realistic scenario). However since the token is expired after initial use, the attack cannot be sustained once the session is required to be renewed.

Figures 6(a-e) show our simulation results. Fig 6(a) and 6(b) show that the availability of the system is much higher in all the overlay based cases in comparison to the unicast case. The ODon case in particular maintains a very high availability metric even under DoS attacks and the flash events during the emergency scenario.

Fig 6(c) and 6(d) show the total response times of the system under varying conditions and varying levels of traffic. For lower loads in the system the unicast traffic shows better latency than the overlay based cases. However as the load increases, OverDoSe and ODon show a very consistent value of latency as opposed to the unicast case. SOS suffers from a large overhead introduced by its circuitous routing. ODon again outperforms OverDoSe because of the larger number of concurrent paths used and switching of forwarding nodes before they become overloaded.

Fig 6(e) shows the percentage of completed legitimate session requests under a distributed request flood attack. As can be seen from the figure, OverDoSe suffers from a large number of dropped sessions due to the overloaded server. ODon filters out most of the spurious requests at the overlay level itself maintaining a near perfect session establishment rate.

Our simulations demonstrate the effectiveness of ODon under loaded network conditions and DoS attacks on the system. It performs consistently better than the unprotected cases and comparable work in DoS-resistant overlay networks.

VIII. CONCLUSION

In this paper we considered the problem of creating a secure, highly available network for mission-critical applications. We believe that our work is the first of its kind which deals exclusively with the nuances of the mission-critical application scenario. For ease of deployment and to obtain high availability we designed the system as an overlay network with forwarding nodes. We designed and verified the correctness of the required access verification and session establishment protocols and also evaluate the performance of the system with a combination of simulation and implementation. Our results show that the ODon system is highly effective in meeting the stringent requirements of the application and performs better than comparable work in related areas.

REFERENCES

- [1] K. Carlberg and R. Atkinson, "General requirements for emergency telecommunication service," February 2004.
- [2] P. World, "Feds may build their own internet," available at <http://www.pcworld.com/article/id,65706-page,1/article.html>.
- [3] Wired.com, "Cerf disses bush's patch plan," available at <http://www.wired.com/politics/law/news/2001/12/49095>.
- [4] Y.-C. Hu and P. A., "A survey of secure wireless ad hoc routing," *IEEE Security and Privacy*, vol. 2, no. 3, pp. 28–39, May 2004.
- [5] J. Gao and C. Beard, "Overlay networks to support internet emergency preparedness services," University of Kansas, Tech. Rep., 2004.
- [6] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of 18th ACM SOSP*, Banff, Canada, October 2001.
- [7] K. Gummadi, H. Madhyastha, S. D. Gribble, H. M. Levy, and D. J. Wetherall, "Improving the reliability of internet paths with one-hop source routing," in *Proceedings of 6th USENIX Symposium on Operating systems design and implementation (OSDI)*, San Francisco, CA, USA, December 2004.
- [8] A. Keromytis, V. Misra, and D. Rubenstein, "SOS: An architecture for mitigating DDoS attacks," *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Service Overlay Networks*, vol. 22, no. 1, pp. 176–188, January 2004.
- [9] E. Shi, I. Stoica, D. Andersen, and A. Perrig, "Overdose: A generic ddos protection service using an overlay network," Carnegie Mellon University, Tech. Rep., 2006, available at <http://reports-archive.adm.cs.cmu.edu/anon/2006/CMU-CS-06-114.pdf>.
- [10] M. Liotine, *Mission-Critical Network Planning*, 1st ed. Artech House Publishers, October 2003.
- [11] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pages = 612 - 613, month = November, 1979.
- [12] J. Kurian, H. T. Vu, and A. Kulkarni, "Odon: An on-demand overlay network for mission critical applications," University of Texas at Dallas, Tech. Rep., May 2008.
- [13] T. Hoare, *Communicating Sequential Processes*. Prentice Hall, 2004.
- [14] Formal Systems (Europe) Ltd., "Fdr2 user manual," available at <http://www.fsel.com/documentation/fdr2/html/index.html>.
- [15] G. Love, "Breaking and fixing the needham-schroeder public-key protocol using csp and fdr," *Lecture Notes in Computer Science*, vol. 1055, pp. 147–166, 1996.
- [16] G. Lowe and A. Roscoe, "Using csp to detect errors in the tmn protocol," *IEEE transactions on Software Engineering*, vol. 23, 1997.
- [17] S. Schneider, "Verifying security protocols: an application of csp," in *Proceedings of 25 Years of CSP*, London, England, July 2004.
- [18] P. Ryan and S. Schneider, *Modelling and Analysis of Security Protocols*, 1st ed. Addison-Wesley, August 2001.
- [19] National Research Council (U.S.A.), *The Internet Under Crisis Conditions: Learning from September 11*, 1st ed. National Academies Press, January 2003.