NYMBLE: Providing Anonymity to Users and Blocking Misbehaving Users

Swapnil Chidrawar¹, Raj Undale², Akshay Ghawate³, Krishna Mule⁴, Prof. D. J. BONDE⁵
M.M.I.T, LOHAGAON, PUNE-411015

Abstract— In this paper we present a anonymous network named “NYMBLE”, in this network users can access the network without providing their IP address. Their IP address will be hidden and any user misbehaving in the network will be blocked when a complaint will be received against the user. The system which we aim to develop in this project is the Nymble System Commanded by Desktop Application. The aim of this project is to create a anonymous network. In this network the IP-address of the user will be hidden to provide anonymity and also to block access to misbehaving users in anonymous network.

Here, we present a system “Nymble” in which servers can blacklist the misbehaving users, that is blocking users without compromising their anonymity. There are many anonymous network which allow users to access Internet services privately by using a series of routers for example “TOR”. These routers hide the client’s IP address from the server. The success of these networks however, has been limited by the users and employing this anonymity for abusive purposes such as defacing popular and important Web sites. Web site administrators routinely rely on IP-address blocking for disabling access to misbehaving users, but blocking of users is not practical in a anonymous network, because the website administrators cannot find the user in a anonymous network. The website has to block the entire network, which is not practically possible. To address this problem, we present Nymble, a system in which servers can blacklist a misbehaving user, thereby blocking users without compromising of their anonymity. Our system is providing a solution to different servers a definitions of misbehavior servers can blacklist misbehaving users for whatever reason, and the privacy of blacklisted users is maintained.

Keywords: - Anonymity, fast authentication, unlinkability, auditability.

II. ARCHITECTURE

Fig.1 System Architecture

Our research makes the following different contributions:
1. Anonymous authentication
2. Backward unlinkability
3. Subjective blacklisting
4. Fast authentication speeds
5. Rate-limited anonymous connections
6. Revocation auditability : where users can verify whether they have been blacklisted.

In Nymble, users acquire an ordered collection of nymbles, and a special type of pseudonym, to connect to Websites. Without any extra additional information, and these nymbles are computationally hard to link, therefore using the stream of nymbles simulates anonymous access to services. Any web sites, however, can blacklist users by obtaining a seed for a particular nymble, allowing them to the link future nymbles from the same user, those used before the complaint remain unlinkable.

Servers can therefore blacklist anonymous users without knowledge of their IP addresses while allowing behaving users to connect anonymously. Our system ensure that users are aware of their blacklist status before they present a nymble, and the disconnect immediately if they are blacklisted. Although our any work applies to anonymizing networks in general, we will consider the Tor for purposes of exposition. If in any case any number of anonymizing networks can rely on the same Nymble system, blacklisting the anonymous users regardless of their anonymizing network(s) of choice.
A. Main Modules In NYMBLE

1) Pseudonym Manager:
   The user must first contact the Pseudonym Manager (PM) and demonstrate control over a resource, for IP-address blocking, the user must always connect to the PM directly (i.e., not through a known anonymizing network). We assume the PM has knowledge about Tor Routers, and can ensure that users are communicating with it directly. Pseudonyms are always deterministically chosen based on each controlled resource, we will ensure, that the provided pseudonym is always issued for the same resource. Note that the user does not disclose what the server he or she intends to connect to, and the pseudonyms managers duties are limited in the nymble to mapping IP addresses (or other resources) to pseudonyms. As we know that, the user contacts the PM only once per linkability window (e.g., once a day).

2) Nymble Manager:
   After receiving pseudonym from pseudonym manager the user sends the pseudonym to nymble manager. Nymble manager verifies the pseudonym and after verification provides access rights to the user.
   The nymble manager also updates the blacklist. The blacklist is maintained at the server, when a complaint is received at the server. The server send ticket to nymble manager. The nymble manager checks whether this ticket is already present in the blacklist if it is not present in the blacklist. It is added to the blacklist.

3) User:
   User registers himself to the pseudonym manager and sends the I.P address to the pseudonym manager and receives the pseudonym from him and uses this pseudonym instead of the I.P address. Some users may also misuse this service, so we block any misbehaving user in the nimble network.

4) Server:
   The server provides internet access to the user it even sends the complaint ticket to the “Nymble Manager”. The nymble manager checks whether this ticket is already present in the blacklist if it is not present in the blacklist. It is added to the blacklist.

The above figure shows the linkability window, the nymble tickets are bound to a specific time period. This linkability window is divided into ‘w’ time periods which is further divided into L time periods of duration T. Each user is blacklisted for a single linkability window only. At the end of each linkability window the blacklist, encryption and decryption keys are updated. This is done to make the network synchronous. And also it ensures forgiveness of misbehaving users after certain period of time, so that the misbehaving users will be allowed to access the service and if they do misbehave again they are blocked again. The blacklist is also updated after each time duration t.

Fig. 1 System Flow

Fig. 3 The life cycle of a misbehaving user

Fig. 4 To generate random tickets
In above figure, to generate ticket for multiple users a concept of seed and nymble is used. Nymble, which is a pseudorandom number, and it serves as an identifier for a particular time period. This nymbles are presented as part of a nymble ticket, as described next. When a new ticket is to generated it is linked to seed; A function $f$ is applied on the current seed and the next seed is generated, and the new nymble is linked to the new seed generated by the function $f$. The seed for the next time period (seednext) is computed from the seed for the current time period (seedcurrent) as seednext=f(seedcur).

![Diagram of nymble ticket](image)

**Fig 5. To receive updated blacklist**

In above figure a concept of daisy and target is used for complaint, and for each complaint the Nymble Manager generates a new random daisy. And this daisy points to the target. So each time the user can receive a updated blacklist and the old blacklist can be discarded.

**B. Algorithm:**

1) System setup.

2) Server Registration.

3) User Registration.

4) Nymble Connection Establishment:
   - Blacklist Validation
   - Ticket Examination.

5) Blacklist Update

6) Periodic Update

**Abbreviation:**

PM-Pseudonym Manager
NM-Nymble Manager
MAC-Message authentication code

---

1) **System Setup**

\[
\begin{align*}
1: & \quad \text{macKey}_{NP} := \text{Mac.KeyGen}() \\
2: & \quad \text{macKey}_N := \text{Mac.KeyGen}() \\
3: & \quad \text{seedKey}_N := \text{Mac.KeyGen}() \\
4: & \quad (\text{encKey}_N, \text{decKey}_N) := \text{Enc.KeyGen}() \\
5: & \quad (\text{signKey}_N, \text{verKey}_N) := \text{Sig.KeyGen}() \\
6: & \quad \text{keys} := (\text{macKey}_{NP}, \text{macKey}_N, \text{seedKey}_N, \\
7: & \quad \quad \text{encKey}_N, \text{decKey}_N, \text{signKey}_N, \text{verKey}_N) \\
8: & \quad \text{nmEntries} := 0 \\
9: & \quad \text{return } \text{nmState} := (\text{keys}, \text{nmEntries})
\end{align*}
\]

2) **Server Registration**

**Input:** $(sid, t, w) \in H \times \mathbb{N}^2$

**Persistent state:** $\text{nmState} \in S_N$

**Output:** $\text{serState} \in S_S$

\[
\begin{align*}
1: & \quad (\text{keys}, \text{nmEntries}) := \text{nmState} \\
2: & \quad \text{macKey}_{NS} := \text{Mac.KeyGen}() \\
3: & \quad \text{daisy}_L \in H \\
4: & \quad \text{nmEntries} := \text{nmEntries} || (sid, \text{macKey}_{NS}, \text{daisy}_L, t) \\
5: & \quad \text{nmState} := (\text{keys}, \text{nmEntries})' \\
6: & \quad \text{target} := h^{L-t+3} (\text{daisy}_L) \\
7: & \quad \text{blist} := 0 \\
8: & \quad \text{cert} := \text{NMSignBL}_\text{nmState} (sid, t, w, \text{target}, \text{blist}) \\
9: & \quad \text{serState} := (sid, \text{macKey}_{NS}, \text{blist}, \text{cert}, 0, 0, 0, t) \\
10: & \quad \text{return } \text{serState}
\end{align*}
\]

3) **User Registration**

**Input:** $(uid, w) \in H \times \mathbb{N}$

**Persistent state:** $\text{pmState} \in S_P$

**Output:** $\text{nym} \in P$

\[
\begin{align*}
1: & \quad \text{Extract nymKey}_{P}, \text{macKey}_{NP} \text{ from pmState} \\
2: & \quad \text{nym} := \text{MA.Mac}(uid || w, \text{nymKey}_{P}) \\
3: & \quad \text{mac} := \text{MA.Mac}(\text{nym} || w, \text{macKey}_{NP}) \\
4: & \quad \text{return } \text{nym} := (\text{nym}, \text{mac})
\end{align*}
\]
4) Nymble Connection Establishment

i. Blacklist Validation

Input: \((s_id, t, w, b\text{list}, c\text{ert}) \in H \times \mathbb{N}^2 \times B_n \times C, n \in \mathbb{N}_0\)

Output: \(b \in \{\text{true, false}\}\)

1. \((t_d, d\text{aisy}, t_\text{e}, m\text{ac}, \text{sig}) := c\text{ert}\)
2. if \(t_d \neq t \lor t_d < t_\text{e}\) then
3. return false
4. \(\text{target} := h^{t_d-t_\text{e}}(d\text{aisy})\)
5. \(\text{content} := s_id||t_\text{e}||w||\text{target}||\text{b\text{list}}\)
6. return \(\text{Sig.Verify(content, sig, verKey_N)}\)

ii. Ticket Examination.

Input: \(\text{ticket} \in T\)

Persistent state: \(\text{srvState} \in S_S\)

Output: \(b \in \{\text{true, false}\}\)

1. Extract \(\text{linking-tokens from srvState}\)
2. \((\cdot, \text{nymble,}\cdot\cdot\cdot) := \text{ticket}\)
3. for all \(i = 1\) to \(|\text{linking-tokens}|\) do
4. if \((\cdot, \text{nymble}) = \text{linking-tokens}[i]\) then
5. return true
6. return false

5) Blacklist Update

Input: \((s_id, t, w, \text{b\text{list}}, \text{c\text{omplet\text{-}tickets})} \in H \times \mathbb{N}^2 \times B_n \times T^m\)

Persistent state: \(\text{nm\text{State}} \in S_N\)

Output: \((\text{b\text{list}}, \text{c\text{ert})} \in B_n \times C\)

1. \((\text{keys, nmEntries)} := \text{nm\text{State}}\)
2. \((\cdot, \text{macKey_N}, \text{seedKey_N}, \cdot) := \text{keys}\)
3. for \(i = 1\) to \(m\) do
4. \((\cdot, \cdot, \text{ctxt}, \cdot, \cdot) := \text{complet\text{-}tickets}[i]\)
5. \(\text{nymble}^*||\text{seed} := \text{Decrypt(ctxt, decKey_N)}\)
6. if \(\text{nymble}^* \in \text{b\text{list}}\) then
7. \(\text{b\text{list}}[i] \in E \ H\)
8. else
9. \(\text{b\text{list}}[i] := \text{nymble}^*\)
10. \(\text{daisy}_L \in R \ H\)
11. \(\text{target}' := h^{t_d-t_\text{e}}(\text{daisy}_L')\)
12. \(\text{cert}' := \text{NM\text{Sign}\text{BL}(sid, t, w, target', bl\text{ist}[i]||bl\text{ist}')}\)
13. Replace \(\text{daisy}_L\) and \(\text{t\text{e}\text{review}}\) in \(\text{nmEntries[sid]}\) in \(\text{nm\text{State}}\) with \(\text{daisy}_L'\) and \(t\), respectively
14. return \((\text{b\text{list}}, \text{c\text{ert}')})\)

6) Periodic Update

Persistent state: \(\text{srvState} \in S_S\)

1. Extract \(\text{linking-tokens from srvState}\)
2. for all \(i = 1\) to \(|\text{linking-tokens}|\) do
3. \((\text{seed}, \text{nymble}) := \text{linking-tokens}[i]\)
4. \(\text{seed}' := f(\text{seed}); \text{nymble}' := g(\text{seed})\)
5. \(\text{tokens}[i] := (\text{seed}', \text{nymble}')\)
6. Replace \(\text{linking\text{-}tokens}\) in \(\text{srv\text{State}}\) with \(\text{tokens}\)
7. Replace \(\text{seen\text{-}tickets}\) in \(\text{srv\text{State}}\) with \(\)
C. Advantages
3. Practical Performance.
5. Backward Unlinkability.
6. Rate Limited anonymous connection.

D. Disadvantages
1. Cannot get the actual contents of the message.
2. Nymble can’t block any user from the network until it receives any complaint.

III. Conclusion
Proposed and built a comprehensive credential system called Nymble, it can be used to add with a layer of accountability to any publicly known Anonymizing Network. The servers can blacklist misbehaving users while maintaining a way that is practical and the efficient, sensitive to the needs of both users and services.

The propose solution will increase the mainstream acceptance by several services because of users who abuse their anonymity.

REFERENCES