2014 Tenth International Conference on Signal-Image Technology & Internet-Based Systems

Connected Components and Credential Hopping in Authentication Graphs

Aric Hagberg and Nathan Lemons
Center for Nonlinear Studies
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545

Alex Kent and Joshua Neil
Advanced Computing Solutions
Los Alamos National Laboratory
Los Alamos, NM 87545

***Abstract*—Modern enterprise computer networks rely on centrally managed authentication schemes that allow users to easily communicate access credentials to many computer systems and applications. The authentication events typically consist of a user connecting to a computer with an authorized credential. These credentials are often cached on the application servers which creates a risk that they may be stolen and used to hop between computers in the network.**

We examine computer network risk associated with credential hopping by creating and studying the structure of the *authentication graph*, a bipartite graph built from authentication events. We assume that an authentication graph with many short paths between computers represents a network that is more vulnerable to such attacks. Under this natural assumption, we use a measure of graph connectivity, namely the size of the largest connected component, to give a quantitative indicator of the network's susceptibility to such attacks. Motivated by graph theoretical results for component sizes in random intersection graphs, we propose a mitigation strategy, and perform experiments simulating an implementation using data from a large enterprise network. The results lead to realistic, actionable risk reduction strategies.

To facilitate continued research opportunities we are also providing our authentication bipartite graph data set spanning 9 months and 708 million time-series edge records.

I. Introduction

Centrally managed user authentication across an organization's network is a primary element of modern enterprise computer environments. Within these networked environments, users often authenticate to many computer systems and applications throughout the network with credentials that allow access to various shared resources such as email servers, file storage systems, printers, corporate web sites, and much more. Typically the credentials are cached on these resources to make repeated access convenient. Unfortunately this convenience introduces the risk of credential misuse: it is possible to steal credentials for use in accessing unauthorized network resources [1], [2]. Given the importance and ubiquity of centralized authentication, quantifying the network-wide risk of this exploitation is critical [3].

We approach this problem by creating a graph of network authentication events. The authentication activity implies relationships between computers and computer users that can be naturally and efficiently represented as a bipartite graph [4]. We find that this graph representation, the bipartite "authentication graph" defined in Section II-C, gives insight into the aggregated relationships between users and computers and serves as a platform for objective risk analytics based on graph attributes.

More specifically, representing user network authentication activity as a set of relationships between users and computers is an effective method to examine the potential risk associated with the two interdependent activities that we define in combination as *credential hopping*. The first activity is the inappropriate use of another user's credentials by an attacker on a computer within the network. The second activity is the inappropriate use of a stolen credential to move from one computer to another. The bipartite representation captures both of these activities across the entire user and computer populations of a network and enables the identification of both inappropriate credential hopping opportunities and mitigation approaches.

In this paper we use real data from a large enterprise computer network model and analyze the credential hopping risk. We study the effect of varying credential cache sizes on the component structure and connectivity of the authentication graph. We find that there is a surprising difference in the connectivity when varying cache size limits. We believe that these results can objectively improve the security of enterprise networks since constrained cache sizes across a set of computers has not previously been considered as a potential mitigation to credential hopping.

II. Authentication graphs

A. Authentication systems

Centralized authentication management and control is a mainstay of modern information technology (IT) administration and is expected in the IT infrastructure of most large organizations [5]. Network authentication is typically a combination of two technologies, Microsoft NTLM authentication [6] and MIT Kerberos [7] authentication. These authentication methods are well integrated into most existing desktop and server-oriented operating systems. Both of these methods employ the use of cached credentials, which are exploitable for re-use (theft) and enable the very real risk of credential exploitation within a network by an adversary [1], [2].

Empirical examination within the Microsoft Windows operating system shows the credential caching policy to be diverse given different scenarios and credential types [8]. For example,

978-1-4799-7978-3/14 $31.00 © 2014 IEEE
DOI 10.1109/SITIS.2014.95

TABLE I. EXAMPLE 10 LINES FROM OUR 708 MILLION LINE DATA SET. THE DATA CONSISTS OF LINES WITH A TIME LABEL (SECONDS STARTING AT EPOCH 1), USER LABEL, AND COMPUTER LABEL REPRESENTING AN AUTHENTICATION EVENT OF THE USER ON A COMPUTER.

time,user,computer
1,U1,C1
1,U1,C2
2,U2,C3
3,U3,C4
6,U4,C5
7,U4,C5
7,U5,C6
8,U6,C7
11,U7,C8
12,U8,C9

Kerberos credentials for a user are cached only for the duration a user is logged into a computer and are only valid for a finite length of time (the default is 10 hours). Originating Kerberos credentials (ticket-granting-tickets) can be renewed while still valid for even longer periods of time (the default is one week). At any given time the number of cached Kerberos credentials on computer is the number of users currently logged in.

In contrast, NTLM authentication credentials (often referred to as NTLM *hashes*, since they are a simple cryptographic hash transformation of the password) are only time constrained by a user changing the originating password. NTLM credentials often exist after users have logged into and then logged out of a computer, particularly in interactive sessions (a graphical interface is used). These credentials persist on the computer until it is turned off or restarted and are only count constrained by the total number of users who have accessed the computer. As a result NTLM credentials are a primary target of credential hopping adversaries. Finally, a third type of cached credential exists that validates authentication events (logins) for users who have previously logged into that computer but cannot currently access the central authentication server. This final type of cached credential is worth mentioning because it is a distinctly different from the previous two and cannot be replayed for credential hopping purposes but is often confused as being associated with NTLM credentials in existing literature [8].

B. Network authentication data

As the basis of exploration for an enterprise-sized centralized authentication system, we analyzed comprehensive authentication data [9] from the enterprise computer network at Los Alamos National Laboratory (LANL). This data comes from two primary sources: the event logs of central authentication servers (the Windows Active Directory servers) and the distributed authentication logs of approximately 22,000 networked computers (desktops and servers).

Our authentication data set represents 9 months of contiguous authentication activity. More specifically, the data set contains 708,304,516 time-ordered, successful user to computer authentication events representing 11,362 users and 22,284 computers. The data set consists of delimited tuples; each contains a time label with one second resolution, a de-identified user, and a de-identified computer as shown in Table I. Note that the data set does *not* contain failed authentication events (e.g., the user failed to successfully login to a computer).

This data set originated from approximately 1.9 billion successful authentication events representing both login events to individual computers by users present at the machine, as well as cross network authentication events from one computer to another by users. The latter case creates two user to computer authentication events with the same time label, one for each computer involved in the lateral authentication movement. Time labels are recorded at the second resolution and we have removed duplicate user to computer authentication events having the same time label. Duplicates may have existed due to multiple legitimate authentication events in the same second. More often, duplicates existed due to the fact that an authentication event can be recorded from up to three locations: an Active Directory server, the originating computer, and the destination computer (in the case of a lateral authentication). Nonetheless, an unquantified number of duplicates may remain in the data for the same authentication event across different time labels due to inconsistent clocks from the differing log event sources. However, we consider these potential duplicates to be generally rare and to be bounded by 600 seconds between any two duplicates due to the time skew constraints of Kerberos.[1]

Additionally, the data set has had all user authentication events to the Active Directory servers removed, since every user continuously authenticates to these servers for a variety of automated housekeeping purposes and otherwise overwhelms the usefulness of the data set. We have also removed all computer account activity (user names ending with the special "$" designation) and only consider human-driven and automated service user accounts.

Finally, we note that users with authentication events to a large number of computers, particularly over short periods of time, often correspond to various automated service accounts used for system monitoring, configuration management, and automated backups; user U12 is one such example within the data set.

C. Graph structure of authentication

We want to explore the authentication data to assess the ease with which an attacker could move through an enterprise network using credential hopping. From Table I it is clear that the data can be naturally viewed as a graph (or graphs if we restrict to intervals of interest) with time labels on the edges. These graphs will all be bipartite; every edge is of the form (u,c) where u represents a user credential and c (one of) the computers on which the credential has been used. This creates an affiliation network [10] between users and computers on the basis of authentication activity, allowing us to explore the relationship and significance between user credentials over a specific set of networked computers over time. A simplified example of how raw authentication events are transformed to the bipartite graph is shown in Figure 1. Note that the transformation from raw authentication log events into user-to-computer authentication events has already been computed for our data set.

[1] Kerberos ticket requests from computers to the Active Directory servers must contain a client generated time label and that time label must be within 5 minutes of the server's assumed clock. If it is not, the client adjusts its system clock appropriately.

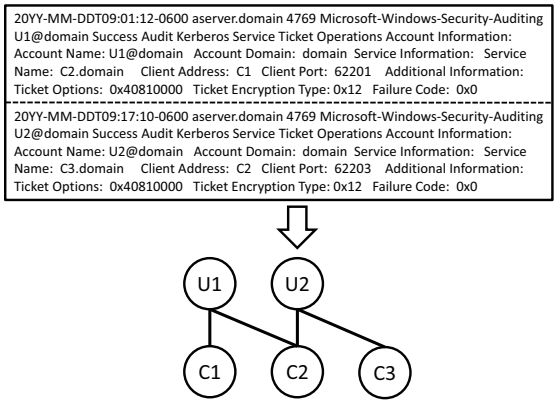

Fig. 1. Users and computers in a simplified bipartite authentication graph from authentication log events. In this example there are 2 users (U1, U2) and 3 computers (C1, C2, C3). Note the potential path using user U1's credentials to get to computer C2 where user U2's credentials could be stolen thus allowing access to C3.

For our analysis, we chose a representative 48 hour window from our longer data set described above. This subset, consisting of two workday periods, is long enough to show our main results. Though results from other days (especially non-workdays) might be quantitatively different, the qualitative results are the same. Figure 2 shows the number of edges (authentication events) and vertices (both computers and users) over this 48 hour period.

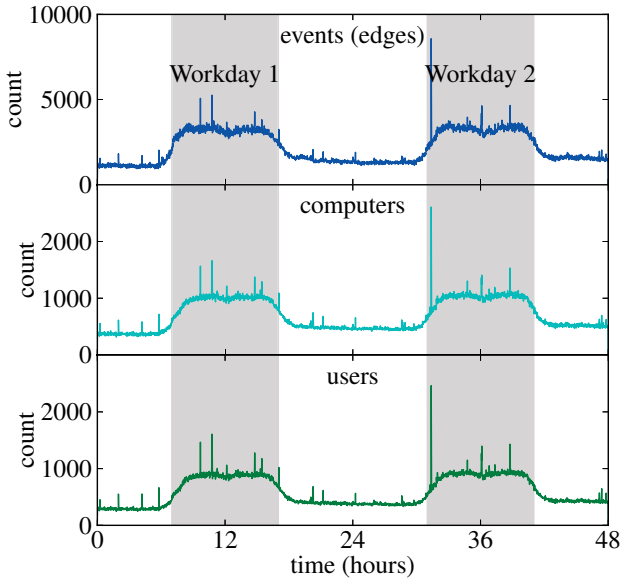

Fig. 2. Time series of events for the first two days of the authentication data set. The data shown are counts of events, computers, and users, recorded in one minute intervals. There is a strong diurnal pattern matching the typical workday. Data from shaded regions labeled "Workday 1" and "Workday 2" will be used in following experiments and analysis. The sharp peak in Workday 2 is due to scheduled, automated activity including backing up to servers and desktop configuration management.

TABLE II. GRAPH STATISTICS FOR THE WORKDAY 1 AND WORKDAY 2 DATA SUBSETS.

	Workday 1	Workday 2
Events	1,879,100	1,923,277
Unique Edges	39,842	40,092
Users	8,272	8,433
Computers	9,857	10,060
Median user degree	4	4
Mean user degree	4.82	4.75
Median computer degree	1	1
Mean computer degree	4.02	3.98

Formally we define an *authentication graph* from a triple $B_I = (U, C, E_I)$; the graph is then defined on the users, U and computers C with edges (u, c) induced from the subset of authentication events E_I indexed by an arbitrary set I. In what follows, we will always consider simple graphs (i.e. graphs without multiple edges) so we will restrict our attention to index sets for which the subsets E_I of data contain each possible pair (u, c) at most once.

As a simple example, suppose we want to build the authentication graph consisting of all authentication events in a 10 hour period. The authentication graph then consists of all edges (u, c) for which user u authenticated (at least once) on computer c within the given 10 hour interval. Formally we could pick E_I to be those events in the 10 hour period in which each user-computer pair first appears, though any set of events that includes each edge (u, c) and avoids multiple instances of the same edge would produce the same graph. Table II gives some basic statistics of the graphs built from the Workday 1 and Workday 2 time periods. Note that the mean and median user and computer degrees[2] are not in close agreement, indicating data skew. In fact, both the distribution of user degree and computer degree are skewed (see Figures 3 and 4) with a few high degree and many low degree values. The skewness is more marked in the computer degree distribution.

We are interested in graph metrics that can asses the risk of credential hopping within the network. All other things being equal, denser graphs imply more and shorter paths between vertices in the graph. Shorter average path lengths between computers means an attacker will more quickly traverse from a computer of little interest to desired computers using credential hopping. More importantly, if an attacker is forced to steal several credentials, going through several network hops to reach computers of interest, they are more likely to be detected. In particular, tools such as PathScan [11], which look for anomalously correlated network traffic to find attackers, would have a higher signal-to-noise ratio if credential stealing attackers were forced to go through more network hops.

Perhaps the simplest question regarding graph connectivity is the following: which vertices (both computers and users) are connected to each other through paths? Using standard graph terminology, the question becomes: What are the connected components of the graph? From a security point of view, *the ideal situation is when the authentication graph has many components of small size*. This would imply that an attacker on part of the network would only be able to reach a small subset of the computers on the network through credential hopping. We find the size of the largest connected component to be

[2]The degree of a user is the number of computers to which it is connected.

a simple graph-connectivity measure which has reasonable implications for network vulnerability: in the following we concentrate on this metric.

It is clear from Figure 2 that the authentication events exhibit strong diurnal patterns: most activity falls between 7AM and 5PM. There is also significantly less activity on weekends and holidays. The time series dynamics are also driven by the fact that the LANL network is set up so that credentials expire in 10 hours when Kerberos is used. Unfortunately, there is no expiration associated with NTLM credentials. Nonetheless, we find it relevant to look at those authentication graphs where E_I is an authentication events set containing 10 hour periods and ending at 5PM. These graphs appropriately represent the enterprise workday and are the densest: they are the worst case in terms of credential hopping risk. These 10 hour graphs also simplify analysis by ignoring the temporal aspect of data and focusing on one static graph for each workday. In this way we do not have to worry about the diurnal patterns in the data. We note however, that one could define authentication graphs at each moment in time (at the one second resolution for our data set) and consider the whole data set as a large temporal graph. We believe such approaches could be extremely fruitful, but leave such analyses for the future.

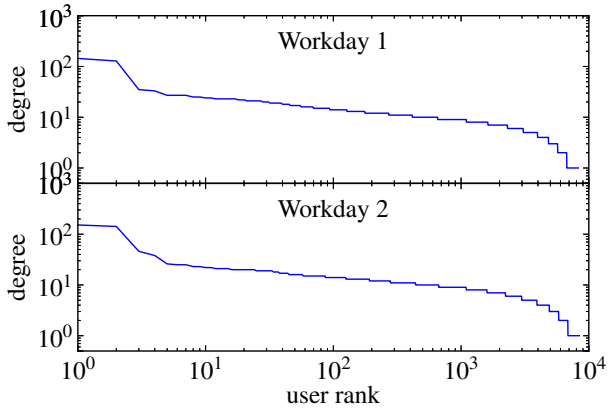

Fig. 3. The number of computers that users authenticate on (the degree of a user) in user rank order highest to lowest. For these two typical 10 hour workdays the majority of users connect to approximately 10 computers each day. Only two users connect to more than 100 computers.

III. Random intersection graphs

We have seen that representing the authentication data as a graph can provide insight into the vulnerability of the underlying enterprise network to credential hopping attacks. To further analyze the structure of the graphs, and to seek mitigation strategies, we present a basic random graph model and connect it to the data. We identify simple mechanisms, which, when implemented in an enterprise network, would result in a reduction in the vulnerability of the network, as measured by our metric of largest connected component size.

We use a random graph model to explain general phenomena observable in the data, as opposed to finding and fitting a high-fidelity model. As we will see, simple models provide insight into underlying structure of the real data. We show that the model gives insight about the data even when the model and the data diverge. In general, the benefit in searching for the simplest model is added clarity and understanding. A high-fidelity model for the authentication graph is thus beyond the scope of this paper and we leave such endeavors for future work.

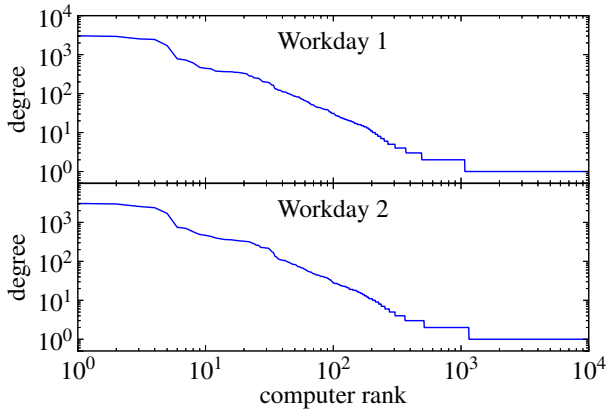

Fig. 4. The number of users authenticated on each computer (the degree of a computer) in computer rank order highest to lowest. For these two typical 10 hour workdays the majority of computers (approximately 9000 of 10000) have only one authenticated users. A few computers have more than 1000 authenticated users.

A simple, natural model for bipartite networks is the random intersection graph. Random intersection graphs were introduced by Karoński, Scheinerman and Singer-Cohen [12], [13] where the graph $G(n,m,p)$ is defined. This is a random bipartite graph B with parts of size n and m, called the vertices and attributes, respectively[3]. Each of the nm possible edges in the graph appears independently with probability p. In essence, this is the bipartite version of the Erdős-Rényi random graphs. A similar model, denoted $G(n,m,d)$, where d is a positive integer, was introduced in [14]. This models also defines a bipartite graph B on n vertices and m attributes. Here each vertex is connected randomly to exactly d attributes (the d attributes are chosen uniformly at random). In the following we will use the model $G(n,m,d)$ to compare with our data.

To describe the relevant mathematical results on random intersection graphs, we will make use of the following standard notations. A graph property P_n holds *asymptotically almost surely* if $\mathbb{P}[P_n] \to 1$ as $n \to \infty$. Let f and g be functions of n. We say $f = O(g)$ if there exists a constant C such that for all n, $f(n) \leq Cg(n)$. We say $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$ both hold.

A. The model $\mathbf{G(n,m,d)}$

In the model $G(n,m,d)$, all the vertices have fixed degree d (they are connected randomly to d attributes) while the distri-

[3] In many applications, it is convenient to view the so called "intersection graph" of the bipartite graph. This is a graph only on the vertices of the bipartite graph; two vertices are connected in the intersection graph if they share at least one common attribute in the bipartite graph B. Hence the name random intersection graph.

bution of the attribute degrees is $\text{Bin}(n, d/m)$. In the asymptotic limit the distribution of attribute degrees is $Poisson(dn/m)$.

Similar to the model $G(n,p)$, the remarkable phenomenon of the phase transition in the size of the largest component is also observed in this model. Here the relevant statistic is

$$\frac{d(d-1)n}{m},$$

a proxy for the average degree in the intersection graph. The result is formalized in the following theorem [15].

Theorem 1. *Let c be fixed and let $d = d(n) \geq 2$ and $m = m(n)$ be functions of n. Let $G(n,m,d)$ be a random intersection graph such that*

$$\frac{d(d-1)n}{m} \to c.$$

- *If $c < 1$ then asymptotically almost surely the largest component in G has size at most $O(\log n)$.*
- *If $c > 1$ then asymptotically almost surely the largest component in G has size $\Theta(n)$ and all other components have size at most $O(\log n)$.*

As an example, consider the random intersection graph with parameters $n = 1000$ and $m = 10000$. Theorem 1 predicts a phase transition at $d \approx 4$. Figure 5 shows a plot of the largest component in instances of this model for varying parameter d. It is clear that there is a phase transition near $d = 4$ with a large jump in the component size, from approximately 0 at $d = 3$ to about 4000 at $d = 5$, which is consistent with the Theorem 1.

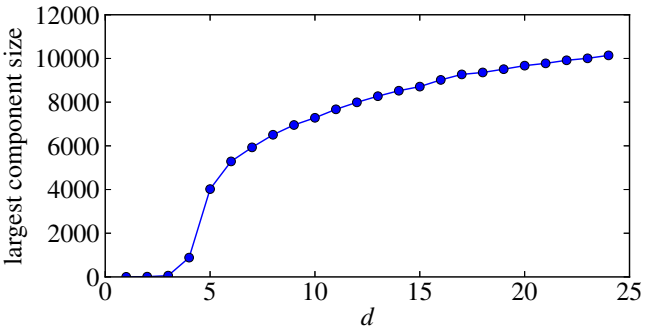

Fig. 5. The size of the largest component in a random intersection graph $G(n,m,d)$ with $n = 1000$ and $m = 10000$. The parameter d varies from 1 to 25. Only one graph was generated for each value of d. The sharp phase transition in the size of the largest component near $d = 4$ is predicted by Theorem 1.

IV. Modifying credential caches

Our goal is to seek credential parameters that may be adjusted to reduce the size of the largest component of the authentication graph and thus reduce the availability for credential hopping. A conceptually simple adjustable parameter that might be used to control the authentication graph is to place a limit on the number of credentials allowed to be stored (cached) on a computer. One might expect that authentication graphs produced under such policies would be similar to the model $G(n,m,d)$ introduced above. Vertices in the random graph represent computers; each is adjacent to exactly d attributes, which represent the user credentials cached on the appropriate computer (e.g. the adjacent computer in the graph).

In the following we use our data set to perform experiments adjusting the cache size parameter d to test the impact. From the model we expect a transition between a mostly disconnected authentication graph for low cache limits to one with a large connected component with high cache limits.

A. Restricting the credential cache size

Using the data sets of 10-hour Workday 1 and Workday 2 authentication events, we tested the effect of limiting the number of user credentials each computer could cache at any moment in time. We built graphs from the two workday periods while restricting the degree of each computer to a fixed value, d, the cache size limit. When a new user authenticated on a computer at the cache size limit, the oldest credential in the cache was then removed and replaced with the new credential.[4] Thus each computer in the graph had degree at most d; we built graphs for each d between 1 and 25.

Figure 6(a) shows the results for the largest component with varying cache size. At $d = 1$ the graph is not connected (and not useful). There is a strong growth as d is increased but there is no apparent phase transition. However, it is clear that by limiting the number of credentials that can be stored on a computer, we can greatly reduce the size of the largest component in the authentication graph. The lack of phase transition can be explained by the fact that the degrees of the user credentials are skew-distributed (even when the computer degrees are bounded) and thus high-degree users connect the authentication graph. Recall that if the data followed the model $G(n,m,d)$ all computers would have degree d; the users degree distribution would be approximately Poisson distributed.

To test the modified hypothesis that the skew degree distributions account for the extra connectivity in the data, we modified our experiment to also limit the number of computers to which any given user could be authenticated at each moment in time. While this may not be practical to implement on an enterprise authentication system, the resulting degree distribution of the users is closer to that of the model $G(n,m,d)$.

The results of this experiment are shown in Figure 6(b); here there is a noticeable phase transition. The transition occurs around the same point as in our random intersection graph example in Figure 5. There, we choose parameters $n = 1000$ and $m = 1000$ to loosely model the structure of the graphs in this last experiment. The parameter m, set to $10,000$, is an approximation of the number of users in the data; the parameter n, set to $1,000$, models the "active" computers in the data. While the data consists of about 10,000 computers, approximately 9,000 out of 10,000 computers have degree 1 (cf. Figure 4). These computers, while contributing to the eventual size of the largest component in the authentication graph, do not influence the evolution of the component size as the parameter d is varied. In deciding which model parameters to use in comparing $G(n,m,d)$ to the experiment, it is natural to only consider the "active" computers as they make up the

[4] When a credential was renewed on the computer, it was moved to the back of the queue.

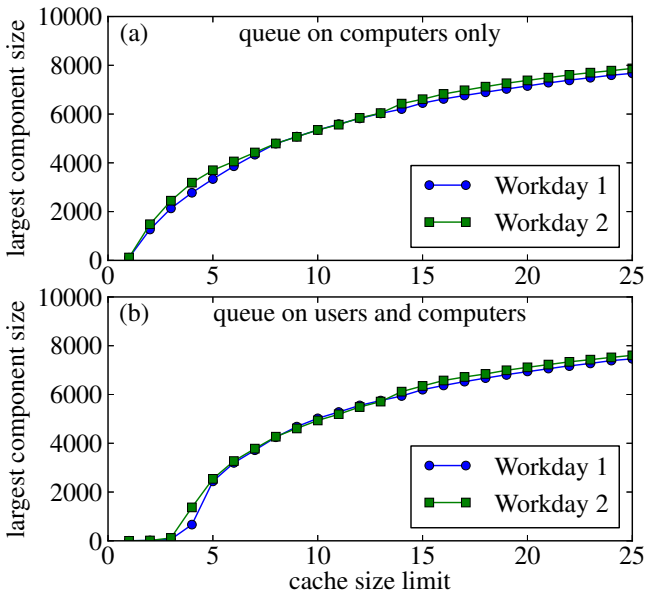

Fig. 6. The effect of setting credential cache limits on computers and users during 10 hour daytime activity periods. Setting a limit on the number of credentials on computers (a) has the effect of reducing the largest component size. But to reduce the size to less than 10% of the unrestricted size would require a cache limit of 2 which is not likely practical. Setting a limit on the number of credentials for both computers and users (b) has a much stronger effect at low cache size limits as is predicted by the random graph model. Reducing the cache size to 4 or less in that case provides a significant reduction in the size of the largest component.

part of the data most like the random graph model. Indeed, the component evolution shown in Figures 6 and 5 is remarkably similar.

Figures 7 and 8 depict the graphs computed in the experiment of Figure 6(b) for $d = 3$ and $d = 5$ respectively. The vertices shown with black circles are the computers and the users are found at the junctions of edges. The largest component in each graph is shown with red vertices marking the computers. The size of the largest component changes dramatically when adjusting the cache limit from $d = 3$ to $d = 5$ but notice that even in the $d = 5$ case most of the rest of the components are still significantly smaller.

In our experiments, we have assumed that the typical user behavior would remain constant under changes to the credential cache size on each computer. This assumption is most likely inaccurate. If actually implemented on the network, this policy would reduce the total number of credentials cached on computers and thus would likely force users to re-authenticate more often. However we do not investigate this dependency here. The assumption of constant behavior may be reasonable when changes to the credential cache limits are relatively modest. We believe the result we have shown will help enable objective decision making in reducing risks associated with credential hopping.

V. Conclusion and Discussion

We used a graph-based approach to modeling authentication data from a large enterprise computer network. By relating the authentication graphs built from this data set to a simple random graph model, we have explained and quantified connectivity phenomena in the graphs. We also show that the skew degree distributions of both the users and computers make the authentication graphs more connected than predicted by the simple model presented here. But even without high-fidelity models we have identified an effective and easily implementable method to limit the number of credentials stored across networked computers. Our experiments on the data show how such policies might affect the connectivity of the authentication graph and how they could be used to control the risk of credential hopping.

The models we used would better represent the data if we remove (or limit) the high degree computers and users. Such an approach, while seemingly ad-hoc might be quite reasonable from the practical point of view. For instance it might be fair to assume that computers which are used by a large number of users need additional security resources devoted to them (e.g. a higher level of monitoring); the majority of computers would not need high-level monitoring but are well modeled collectively by random intersection graphs. Indeed to some extent we have already followed this approach: all data referencing the Active Directory was taken out of the data set as all users associate with the Active Directory Servers.

Our analysis could also be extended to more complex models that may better capture the structure of the data. It would be interesting to compare the authentication graphs with random intersection graphs tuned to achieve the degree distributions observed in the data. The theoretical work of analyzing such models has been done in the case where one of the parts (in the bipartite graph) has tunable degree distribution [16], but not when both parts have a given degree distribution.

However, even with the constraints of our approach, we believe there is practical value in the results shown in this paper for influencing security policy concerning the reduction of credential hopping risk. We have shown that reducing credential cache size to 4 or less is the most effective at reducing the size of the largest component in the authentication graphs and in risk reduction.

The data set used in this study is unique in several ways. We are not aware of any other publicly available data of user authentication events in a large enterprise computer network. Furthermore the data spans a 9 month period, which should capture interesting temporal variations and correlations. Our analysis in this paper is only a beginning to the type of analysis possible. We concentrated on the size of the largest connected components in carefully chosen static graphs selected out of the data. There are many other static graph analyses which could be also be performed: graph partition schemes, measures of graph centrality, clustering coefficients, assortativity coefficients, among others. But an even more interesting direction would be a temporal analysis of the data to seek patterns, correlations, and relevant models. The network science community has recently started to develop a theory of temporal networks [17], but to a certain extent has been limited by access to quality temporal data sets. We hope that providing this data will help inspire the community address these type of questions.

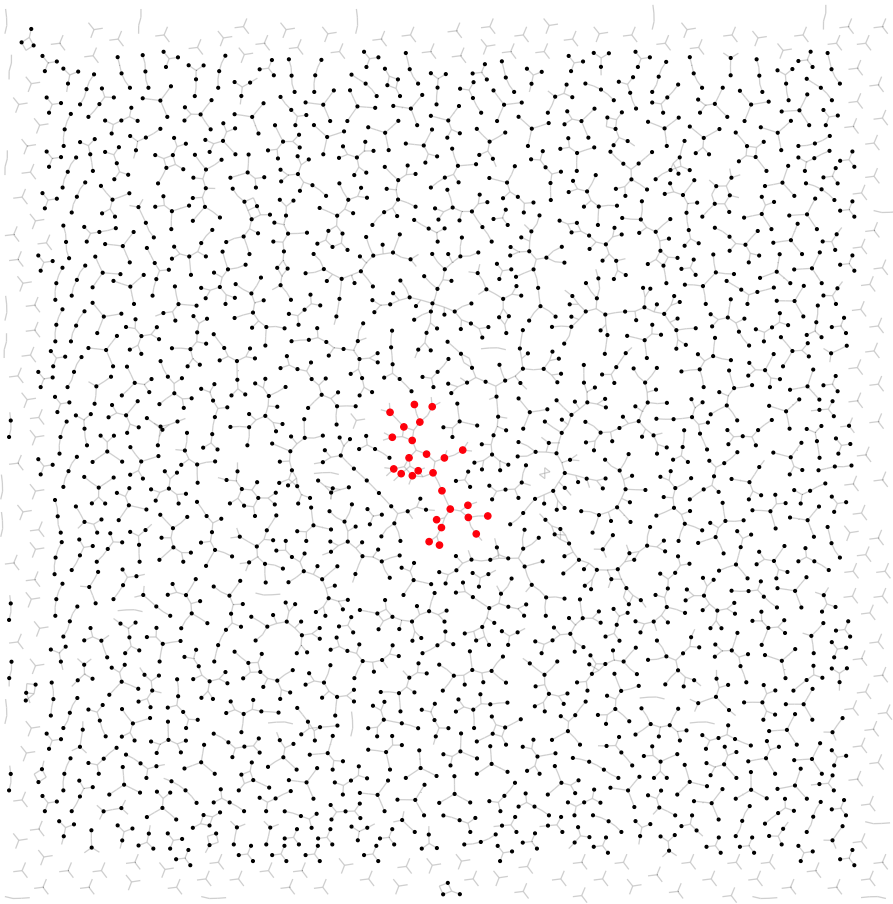

Fig. 7. The authentication graph of Workday 1 with a credential cache limit of 3 users per computer and 3 computers per user. Computers are shown as black circles with those in the largest component colored red. The user vertices are suppressed and appear as junctions in edges between computers. The largest component has 26 computers and 34 users.

VI. Acknowledgments

We would like to thank Pieter Swart and Feng Pan for helpful discussions on data modeling and analysis. This work was supported by the Department of Energy through the LDRD program at Los Alamos National Laboratory.

References

[1] C. Hummel, "Why crack when you can pass the hash?" *SANS Institute InfoSec Reading Room*, 2009.

[2] J. Dunagan, A. X. Zheng, and D. R. Simon, "Heat-ray: Combating identity snowball attacks using machinelearning, combinatorial optimization and attack graphs," in *Proceedings of the ACM SIGOPS 22Nd Symposium on Operating Systems Principles*, ser. SOSP '09. New York, NY, USA: ACM, 2009, pp. 305–320. [Online]. Available: http://doi.acm.org/10.1145/1629575.1629605

[3] J. Johnson and E. Hogan, "A graph analytic metric for mitigating advanced persistent threat," in *Intelligence and Security Informatics, 2013 IEEE International Conference on*, June 2013, pp. 129–133.

[4] A. D. Kent, L. M. Liebrock, and J. Neil, "Authentication graphs: Analyzing user behavior within an enterprise network," *Computers & Security*, 2014.

[5] A. Johnson, K. Dempsey, R. Ross, S. Gupta, and D. Bailey, "NIST guide for security configuration management of information systems," http://csrc.nist.gov/publications/drafts/800-128/draft_sp800-128-ipd.pdf, Mar. 2010.

[6] E. Glass, "The NTLM authentication protocol and security support provider," http://davenport.sourceforge.net/ntlm.html, 2006.

[7] B. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *Communications Magazine, IEEE*, pp. 33–38, 1994.

[8] S. Duckwall and C. Campbell, "Hello, my name is Microsoft and I have a credential problem," in *Blackhat USA 2013 White Papers*, August 2013. [Online]. Available: https://media.blackhat.com/us-13/US-13-Duckwall-Pass-the-Hash-WP.pdf

[9] A. Kent, "Anonymized user-computer authentication associations in time," 2014. [Online]. Available: http://csr.lanl.gov/data

[10] R. L. Breiger, "The duality of persons and groups," *Social Forces*, pp. 181–190, 1974.

[11] J. Neil, C. Hash, A. Brugh, M. Fisk, and C. B. Storlie, "Scan statistics for the online detection of locally anomalous subgraphs," *Technometrics*, vol. 55, no. 4, pp. 403–414, 2013.

[12] M. Karoński, E. R. Scheinerman, and K. B. Singer-Cohen, "On random intersection graphs: the subgraph problem," *Combin. Probab. Comput.*, vol. 8, no. 1-2, pp. 131–159, 1999, recent trends in combinatorics (Mátraháza, 1995). [Online]. Available: http://dx.doi.org/10.1017/S0963548398003459

[13] K. B. Singer-Cohen, "Random intersection graphs," *Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD*,

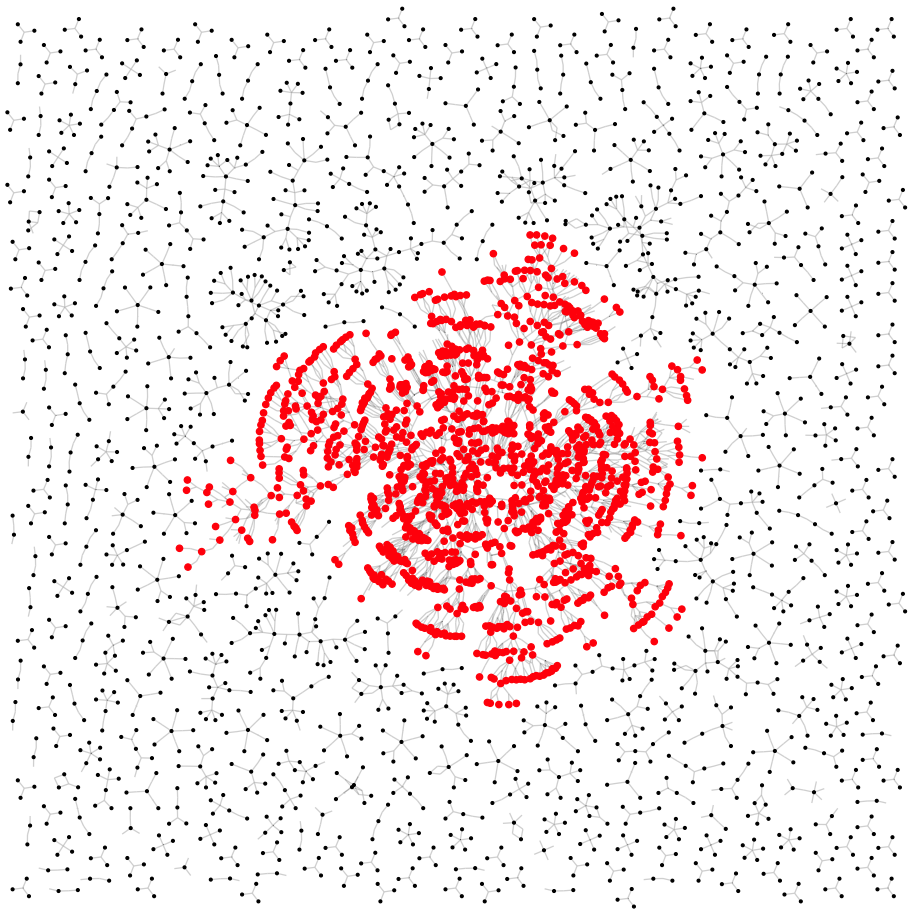

Fig. 8. The authentication graph of Workday 1 with a credential cache limit of 5 users per computer and 5 computers per user. Computers are shown as black circles with those in the largest component colored red. The users vertices are suppressed and appear as junctions in edges between computers. The largest component has 1476 computers and 962 users.

1995.

[14] E. Godehardt and J. Jaworski, "Two models of random intersection graphs for classification," in *Exploratory data analysis in empirical research*, ser. Stud. Classification Data Anal. Knowledge Organ. Berlin: Springer, 2003, pp. 67–81.

[15] K. Rybarczyk, "Diameter, connectivity, and phase transition of the uniform random intersection graph," *Discrete Math.*, vol. 311, no. 17, pp. 1998–2019, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.disc.2011.05.029

[16] M. Bloznelis, "The largest component in an inhomogeneous random intersection graph with clustering," *Electron. J. Combin.*, vol. 17, no. 1, p. R110, 2010.

[17] P. Holme and J. Saramäki, "Temporal networks," *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.